Universidad Carlos III de Madrid

Escuela Politécnica Superior

Ingeniería en Informática

Proyecto Fin de Carrera

Aplicación docente para iPhone Enigmatium (Minijuegos)

Autora: Patricia López Peña

Tutor: Jorge Blasco Alís

Co-tutor: José María de Fuentes García Romero de Tejada

Junio 2011

AGRADECIMIENTOS

Me gustaría agradecer y recordar a todas las personas que han estado a mi lado durante estos siete largos años que ha durado mi andadura universitaria. Han sido años de mucho esfuerzo, sacrificio, de dejar muchas cosas de lado, muchos planes, pero al final, el esfuerzo ha merecido la pena.

El mayor agradecimiento debo dárselo a mi familia, mis padres, mi hermana y mi abuela. Habéis aguantado al pie del cañón todos y cada uno de los días, soportando mis días buenos, pero sobre todos los malos, que han sido muchos. Habéis hecho un gran esfuerzo por ayudarme a conseguir lo que quería y por ayudarme en mi formación. Como tú bien dices Papá, es la mejor herencia que me podías dejar. Agradecerlo por supuesto, a mis otros tres abuelos, que aunque ya no estáis, también compartisteis mucha de esta experiencia conmigo.

Amama, eres la persona más fuerte que he conocido, con un interior tan bueno que irradias paz y tranquilidad a todos los de tu alrededor, me encantaría llegar a ser como tú. Papá, eres mi modelo a seguir en la vida, te admiro profundamente, por todo lo que haces, por cómo eres. Ama, gracias por todos los momentos que me has aguantado, que no han sido pocos, eres una valiente, por todo. Belén, que me has ayudado en tantas cosas, y tú casi sin saberlo, por eso eres tan especial, ¡te adoro!

Álvaro, uno de las últimas personas que he conocido en la carrera y sin embargo el mejor de todos los hallazgos has sido tú. Me has traído equilibrio a mi vida y me has hecho comprender que las cosas también se pueden afrontar de otro modo, desde la tranquilidad. Muchas gracias por todos los consejos, por tu inmenso apoyo incondicional, por aguantarme, por ayudarme y superar estos duros meses conmigo.

Qué duro se hizo aquel primer verano de carrera en el que dudaba si en septiembre podría continuar, pero Ángel se cruzó conmigo en la biblioteca y desde entonces te convertiste en mi "*padre*" en esta universidad. Tú me guiaste desde un principio sobre cómo debía estudiar y tu letra se convirtió en la mía. No te haces una idea de cuánto me ayuste y de cuánto lo valoro y lo valoraré siempre. Después vino Mary, con tu ayuda y consejo en tantas y tantas cosas que no sabría por dónde empezar.

Una vez Ángel, me dijiste que esta era una carrera de fondo, y así ha sido. Una maratón en la que nunca hubiera logrado llegar a la meta sin la ayuda de todos los compañeros, y grandes amigos, que compartieron conmigo momentos amargos, pero también muy dulces. Siempre estaré profundamente agradecida a Andrés, Miguel, Pepe y José-Moncas (mi grupo *The Jokers*),

Brais, José (pelos), Aitor, Jesús, Raúl, Martín, Diego, Jorge, Javi, Chus y todos aquellos que se quedaron por el camino, habéis hecho que la estancia en la universidad sea divertida.

A la persona que siempre llevaré conmigo es a Judith, primero mi compañera de prácticas, luego mi compañera de PFC, pero siempre mi amiga. Tú también me has aguantado mucho durante estos meses de proyecto en los que me he venido abajo o sentía que no tenía fuerzas. Me has dado consuelo y me has ayudado a volver a levantarme. Muchísimas gracias por estos meses de trabajo conjunto, ha sido un verdadero placer trabajar contigo. Eres una grandísima profesional y una excelente persona.

Jorge, nuestro tutor, te conocí en las prácticas de una asignatura de la carrera y desde el primer momento que me diste clase, me pareciste un profesor que se preocupaba por sus alumnos, que se dejaba la piel por ayudarles. Por eso busqué hacer un proyecto contigo, ¡y encima de iPhone! Y con mi primera impresión no me equivoqué. Nos has ayudado todos y cada uno de los meses que hemos estado haciendo el proyecto, desde la primera idea loca que se nos vino a la cabeza hasta el último punto de la memoria que me has ha corregido. Se ve que te gusta lo que haces, y eso nos lo has trasmitido. Muchas gracias por todo lo que nos has enseñado y ayudado, y todo sin una queja, sin una bronca o una mala cara, no se podría tener un mejor tutor.

Parecía mentira pero por fin, esta etapa de mi vida se cierra para comenzar una nueva y emocionante aventura. Sé que muchos de vosotros me acompañaréis también en esta nueva etapa.

¡Cómo no agradecéroslo a todos, si vosotros hacéis que mi vida sea posible!

Índice de Contenidos

Índice de Tablas

Índice de Figuras

1. Introducción

En esta sección se describe la motivación del presente proyecto, sus objetivos y una descripción general acerca del presente documento, tanto la estructura como el contenido de cada una de las secciones en las que se divide el documento.

1.1. Motivación

En el mundo de las tecnologías, es fundamental la seguridad en las mismas y la protección de los datos, tanto de los usuarios como de las empresas que proporcionan esas tecnologías.

El estudio y conocimiento de los métodos, protocolos y algoritmos de seguridad que se pueden y deben utilizar en el marco de las tecnologías, es un objetivo fundamental para los estudiantes de informática.

El estudio de los algoritmos de seguridad y criptografía puede resultar complicado. Cada uno de estos algoritmos tiene un número de pasos claros y precisos que se deben seguir. La resolución de cada uno de estos pasos es llevada a cabo mediante métodos puramente matemáticos, lo que hace que sean bastante complejos.

En muchas ocasiones la falta de tiempo, o de motivación hace que muchos alumnos no practiquen lo suficiente, por lo que no consiguen adquirir un conocimiento claro y preciso acerca de los algoritmos criptográficos estudiados.

En los últimos años, se ha visto una mayor expansión del uso de los llamados *teléfonos inteligentes*, como se muestra en el siguiente gráfico (ver *Imagen 1*). En él se puede ver la evolución del mercado del último trimestre del 2010 en comparación con el último trimestre del 2009. En la imagen 1 se puede observar cómo las ventas de estos dispositivos móviles se vieron incrementadas de manera notable.

Proveedor OS	Ventas Q4 2010 (millones $)	% Mercado	Ventas Q4 2010 (millones $)	% Mercado	Crecimiento Q4'10/Q4'09
Google	33.3	32.9%	4.7	8.7	615.1%
Nokia	31.0	30.6%	23.9	44.4%	30.0%
Apple	16.2	16.0%	8.7	16.3%	85.9%
RIM	14.6	14.4%	10.7	20.0%	36.0%
Microsoft	3.1	3.1%	3.9	7.2%	20.3%
Otros	3.0	2.9%	1.8	3.4%	64.8%
TOTAL	**101.2**	**100.0%**	**53.7**	**100.0%**	**88.6%**

Imagen 1 – Mercado de ventas mundiales de *teléfonos inteligentes*. Fuente Canalys [1]

En la *imagen 2* se puede observar la evolución del mercado de la venta de los diferentes dispositivos llamados *teléfonos inteligentes*.

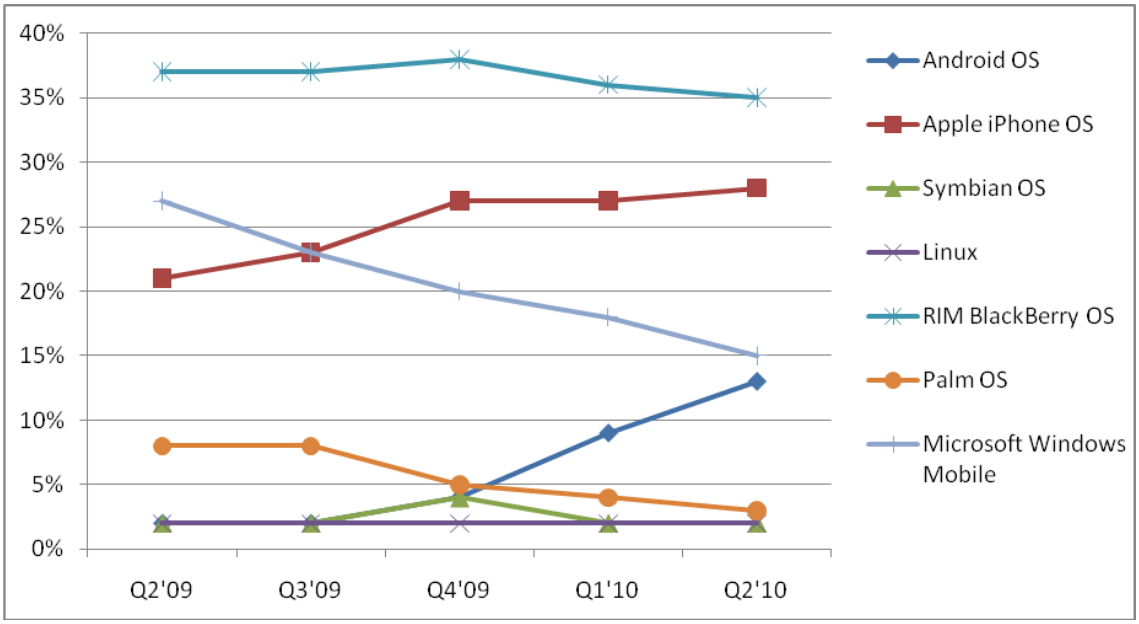


Imagen 2 – Cuota de mercado de sistemas operativos para *teléfonos inteligentes* [2].

De estos y otros estudios que se han podido consultar [3, 4, 5] se puede prever que los *teléfonos inteligentes* continuarán esta tendencia actual a la alza, y por tanto, cada vez más personas dispondrán de un teléfono móvil de última generación.

De entre los diferentes *teléfonos inteligentes* disponibles en la actualidad en el mercado, los principales sistemas operativos para móviles son los siguientes: Android OS (presente en dispositivos como HTC o Samsung), Apple iOS (cuyo único dispositivo es iPhone), Symbian OS (en dispositivos como Nokia, Sony o Ericsson), Linux (presente en dispositivos como Motorola o Samsung) [6], RIM BlackBerry OS (disponible en dispositivos Blackberry), Palm OS (disponible para dispositivos com Sony o HP), Microsoft Windows Mobile (con presencia en dispositivos HP, HTC o Samsung).

De entre todos estos dispositivos que se podrían haber escogido para desarrollar el proyecto, el propio enunciado del mismo, especificaba que la tecnología a utilizar debía ser el iPhone.

De entre las muchas aplicaciones que se pueden encontrar disponibles para *smartphones*, una de ellas son los juegos. Los jóvenes, y por tanto los alumnos, son la parte de la población que más utilizan los juegos en forma de ocio, ya sea en consolas o en móviles.

Dada esta predisposición natural a los juegos por parte de los alumnos, desarrollar un recurso docente en forma de juego, puede motivar a los alumnos a incrementar tanto el nivel como el interés por la asignatura y contenidos de la misma.

Al ser un recurso eminentemente práctico, al alumno adquiere los conocimientos necesarios a través de la diversión y el entretenimiento sin descuidar su formación académica.

1.2. Objetivos

El objetivo que se propone para este proyecto de fin de carrera es implementar una aplicación para iPhone cuya finalidad es ser utilizada como recurso docente en asignaturas con algoritmos criptográficos como contenido temático. De esta manera, se pretende que los alumnos puedan mejorar sus conocimientos a través del uso de la aplicación desarrollada.

Los objetivos a alcanzar mediante el desarrollo del presente proyecto son los siguientes:

Objetivo 1: Contenidos de la aplicación.

La aplicación trata de mejorar los conocimientos de los alumnos en el ámbito de la criptografía, por lo que los contenidos de la misma deben ser aquellos que se estudian, o que se pueden llegar a estudiar y resultan prácticos en la adquisición de conocimientos criptográficos y de seguridad tecnológica.

Objetivo 2: Recurso docente.

Al ser la aplicación un recurso docente, debe ayudar al alumno en la adquisición del conocimiento a través de una metodología diferente a la del estudio clásico. La metodología seguida en la aplicación se basa en la implementación de diferentes algoritmos criptográficos en forma de juegos.

Objetivo 3: Diversidad de juegos

La aplicación incluirá un conjunto de juegos, cada uno de los cuales versará sobre un algoritmo criptográfico diferente. En cada juego el alumno podrá adaptar sus capacidades individuales gracias a los diferentes niveles de dificultad con los que cuenta.

Objetivo 4: Información acerca de cada juego

En cada uno de los juegos se explicará al alumno el algoritmo utilizado, el objetivo del juego, la interacción que tiene que tener el alumno con el dispositivo y las diferentes puntuaciones que puede obtener solucionando el juego.

Aunque se espera que el usuario final sea un alumno de asignaturas de *Seguridad En las Tecnologías de la Información* u otras asignaturas con algoritmos criptográficos como contenido, un usuario ajeno a estos conocimientos podrá usar la aplicación y resolver los juegos. La resolución de los juegos les supondrá un mayor reto, pero tendrán la información disponible con ejemplos acerca de la resolución del juego.

Objetivo 5: Creación de juegos

El usuario no sólo responderá a los retos que la aplicación le proponga, también podrá crear nuevos juegos o retos para el resto de usuarios de esta aplicación.

Objetivo 6: Funcionalidad intuitiva

La aplicación ha sido desarrollada con elementos sencillos de tal manera que no supongan al usuario, tanto si está familiarizado con iPhone como si no lo está, un problema de comprensión.

Objetivo 7: Consulta de los resultados obtenidos

El usuario debe poder acceder en todo momento a los resultados que ha obtenido en los diferentes niveles dentro de cada juego, así como a los puntos totales que se consiguen a lo largo de los diferentes juegos.

Objetivo 8: Personalización de la aplicación

La aplicación debe ser personalizable, de modo que el usuario debe poder introducir datos como el nombre, alias de jugador y correo electrónico. Es-

tos datos son introducidos la primera vez que se acceda a la aplicación, aunque se podrán modificar con posterioridad las veces que se quiera.

1.3. Estructura del documento

En este apartado se describirán las diferentes fases del documento y en qué consiste cada una de ellas.

1. Introducción

Sección en la que se explica el por qué de este proyecto, los objetivos que se persiguen y una explicación acerca de la estructura del documento (parte en la que se encuentra el lector).

2. Estudio de viabilidad del sistema

Sección relativa al estado de la cuestión y las herramientas utilizadas para el desarrollo del proyecto.

3. Gestión del Proyecto

Sección donde se incluye una descripción sobre la metodología de desarrollo utilizada, una planificación tanto inicial como real y los costes asociados a este proyecto. Dentro de los costes del proyecto se incluye un estudio de mercado sobre la venta de *Enigmatium*.

4. Análisis

Sección relativa al análisis del sistema, donde se incluyen los casos de uso o los requisitos software y hardware entre otros.

5. Diseño

Contiene un estudio exhaustivo acerca de la arquitectura que presenta el sistema, así como el modelo de datos utilizado y las interfaces.

6. Implementación y pruebas

Sección donde se especifica la implementación realizada para cumplir los objetivos definidos y las pruebas realizadas. Esta sección de pruebas incluye tanto las pruebas de aceptación como las de usabilidad.

7. Integración con otros proyectos

Sección que contiene una explicación detallada de las partes en común que tiene el presente proyecto con otros proyectos relacionados.

8. Conclusiones y líneas futuras

En esta sección se tratan las conclusiones que se han podido obtener después de la realización del proyecto. Así mismo se incluirán las posibles líneas futuras hacia las cuales sería conveniente orientar el proyecto.

9. Bibliografía, glosario y términos

Sección que contiene información acerca de la bibliografía consultada, el glosario y términos que se considera que deben ser explicados por tratarse de terminología específica o difícil de comprender.

Anexo 1. Manual de Usuario

El manual de usuario se incluye como anexo al presente documento. En él se detalla el funcionamiento de la aplicación para que el usuario final no tenga problema a la hora de utilizarla.

Anexo 2. Test de evaluación por parte del usuario final

El test realizado a los usuarios finales como parte de las pruebas de usabilidad se encuentra como anexo al final del presente documento.

2. Estudio de Viabilidad del Sistema

En esta sección se estudia la viabilidad del sistema mediante un estudio de campo de las aplicaciones existentes anteriores al desarrollo del presente proyecto. Además de describir las herramientas *hardware* y *software* de las que se dispone para el desarrollo del mismo.

2.1. Estado de la cuestión

Este apartado abarca el estudio de las aplicaciones anteriores o similares a la que se va a desarrollar. Para cada aplicación se analiza su funcionalidad, así como las posibles deficiencias o mejoras que se podrían aplicar.

Cabe destacar que el análisis se ha reducido sólo a aplicaciones para iPhone. Esto es así ya que en las especificaciones del proyecto se pedía que la aplicación fuera desarrollada para iPhone, por lo que la búsqueda de ha acotado a este dominio.

El primer análisis se ha hecho en base a las aplicaciones más significativas que se han encontrado en la tienda de App Store de Apple.

Al hacer una búsqueda de juegos relacionados con algoritmos criptográficos, los resultados que arroja App Store son bastante reducidos, no obstante, alguna de las aplicaciones propuestas son bastante interesantes.

Cryptix Lite

El objetivo de este juego es descifrar textos cifrados y descubrir frases célebres de personajes de la historia, como Einstein, Shakespeare, Platón, etc. El descifrado se consigue sustituyendo una letra del texto cifrado por otra.

Los jugadores deben resolver los Puzles tanto de manera vertical como de manera horizontal, con un número mínimo de sustituciones en el menor tiempo posible. Este juego permite al usuario practicar diferentes formas de cifrado ya que cada una de las frases célebres se cifran de un modo diferente, por lo que puede llevarle al jugador días resolverlo.

Así mismo muestra información al usuario acerca del mejor resultado obtenido, el mejor tiempo o el número de Puzles que ha resuelto. El usuario también puede controlar la dificultad del juego mediante la obtención de pistas y ayudas. A continuación se muestran unas capturas de pantalla de las interfaces del juego. Como se puede observar en la *Imagen 3*, las letras de la frase están dispuestas a modo de Puzle. El usuario tiene que descubrir qué letra del texto cifrado se corresponde con otra letra del texto en claro. La sustitución se hace mediante prueba y error, hasta que se consiga obtener el texto en claro.

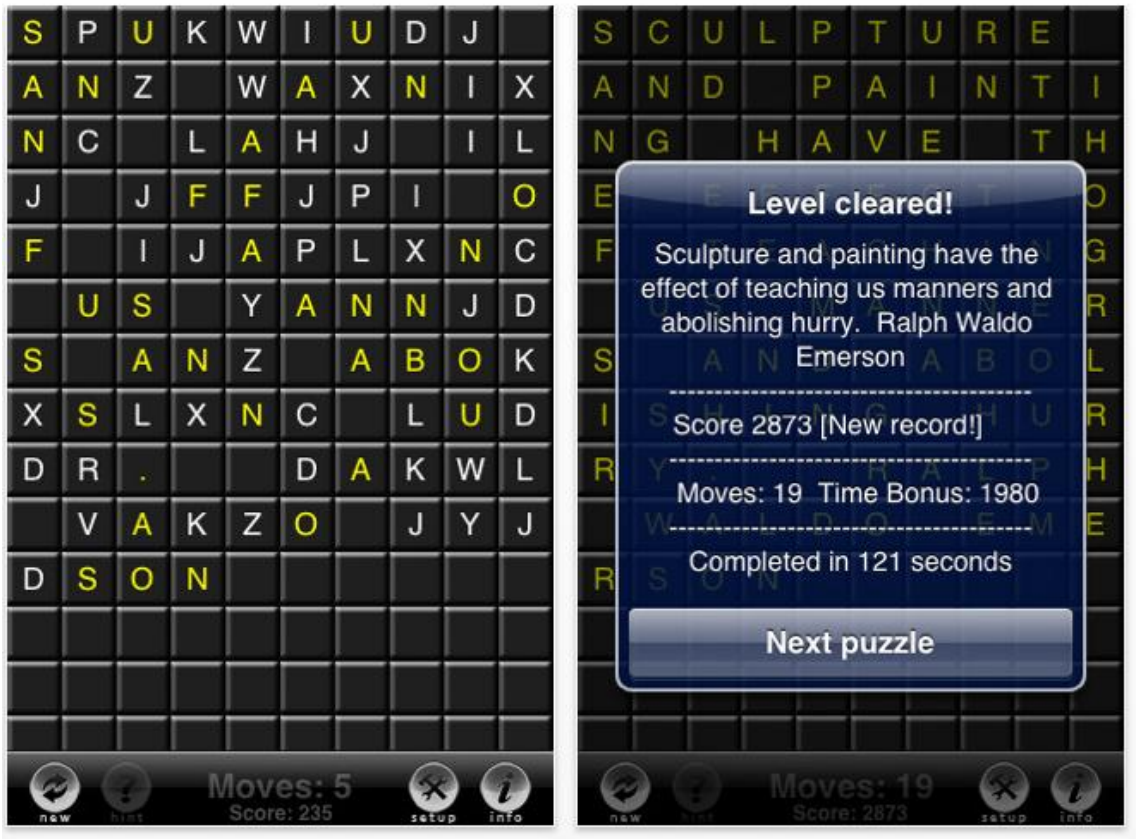


Imagen 3 – Juego Cryptix Lite [10].

En la *Imagen 4* se puede observar las opciones que se proporcionan al usuario, tanto de dificultad como de información acerca de los éxitos obtenidos por parte del usuario.

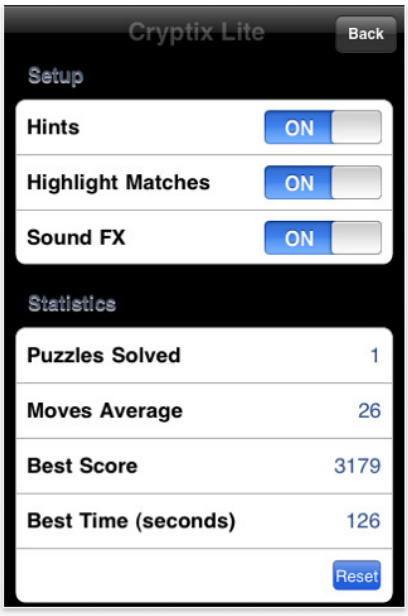


Imagen 4 – Opciones Cryptix Lite [10].

Cryptex 3D Lite

En este juego, el objetivo es resolver un texto que está enrollado en un dispositivo, mediante el desplazamiento de las columnas. Cada columna contiene el abecedario completo.

En la *Imagen 5* se puede observar el aspecto de la pantalla principal del juego, donde se debe resolver un enigma, girando las diferentes columnas del dispositivo, hasta encontrar cuál es el texto en claro. Para encontrar el texto en claro, se debe seguir el mecanismo de prueba y error, es decir, probar hasta que se encuentre la solución. Este juego no presenta ningún tipo de ayuda al usuario por lo que al comenzar a jugar, el usuario se encuentra sin saber qué hacer.

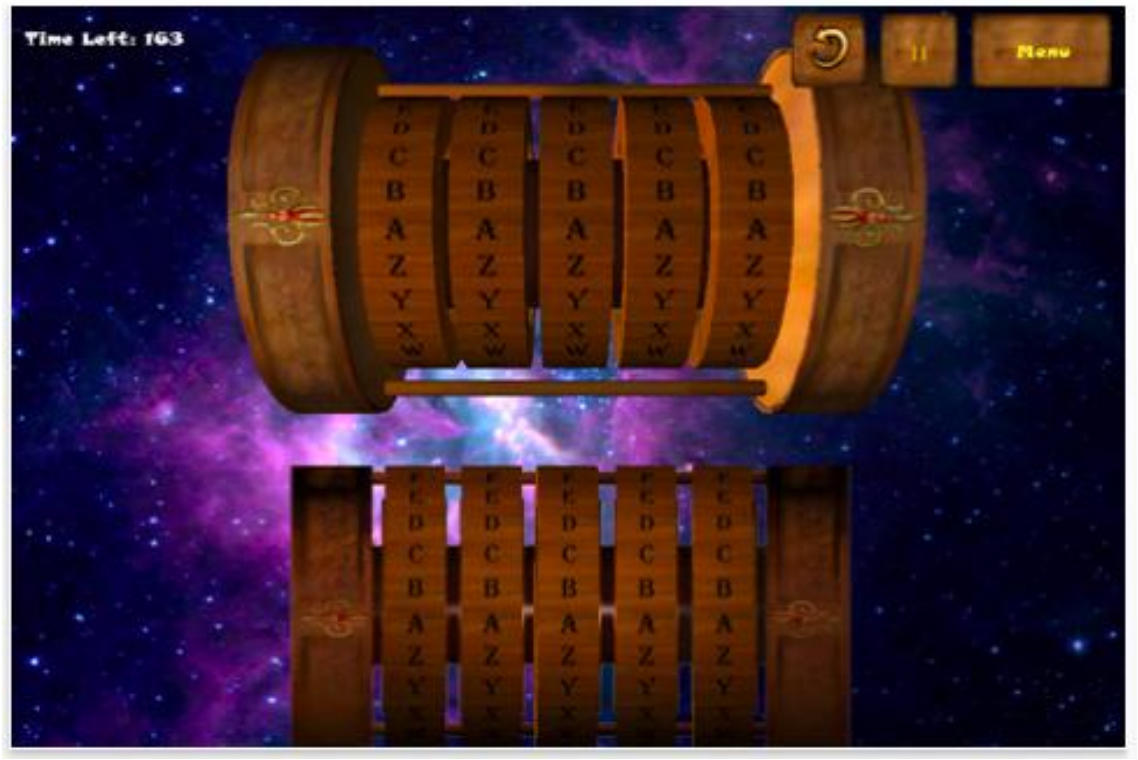


Imagen 5 – Opciones Cryptix Lite [10].

ARG Tool

En este juego se unen varios algoritmos criptográficos, métodos matemáticos, y unos cuantos mecanismos de ayuda para poder resolver textos, como por ejemplo tablas de frecuencias de las letras.

La primera pantalla que aparece es la que se muestra en la *Imagen 6*, donde se puede observar las herramientas que ofrece así como los recursos o referencias que el usuario puede utilizar.

Imagen 6 – Opciones Cryptix Lite [12].

La primera de las herramientas relacionado con la temática del presente proyecto, es *ROT-N*, donde se muestra con un ejemplo un texto cifrado y las posibles soluciones de texto en claro si se aplicara a cada una de las letras un desplazamiento que va desde 1 hasta 25 (están con el alfabeto inglés, no con el castellano, por lo que tienen 26 letras).

En la segunda herramienta *Vigenere/Beaufort* se muestra un texto cifrado y las soluciones si se descifra el texto con una clave dada, tanto por el método de *Vigenere* como por el método de *Beaufort.*

El la herramienta llamada *Transposition Decode* se explica mediante un ejemplo y con varias posibles soluciones, qué tamaño de matriz se debería usar para poder leer el texto que en un primer momento parece cifrado.

La siguiente herramienta *Transposition Decode* muestra justo lo contrario, partiendo de un texto en claro, al colocar el mismo en una matriz de tamaño variable (a elegir por el usuario), cómo quedaría el texto.

En *Base 64*, se muestra un ejemplo de cómo queda un código al volver a aplicarle una codificación en Base 64, así como la decodificación del texto.

Por último, *Substitution* es muy similar a *ROT-N*,se basa en la sustitución de una letra por otra basándose en una palabra dada que actúa de clave.

En la *Imagen 7* se pueden observar dos de estas herramientas mencionadas anteriormente: *ROT-N* y *Substitution* (este segundo basado en el primero).

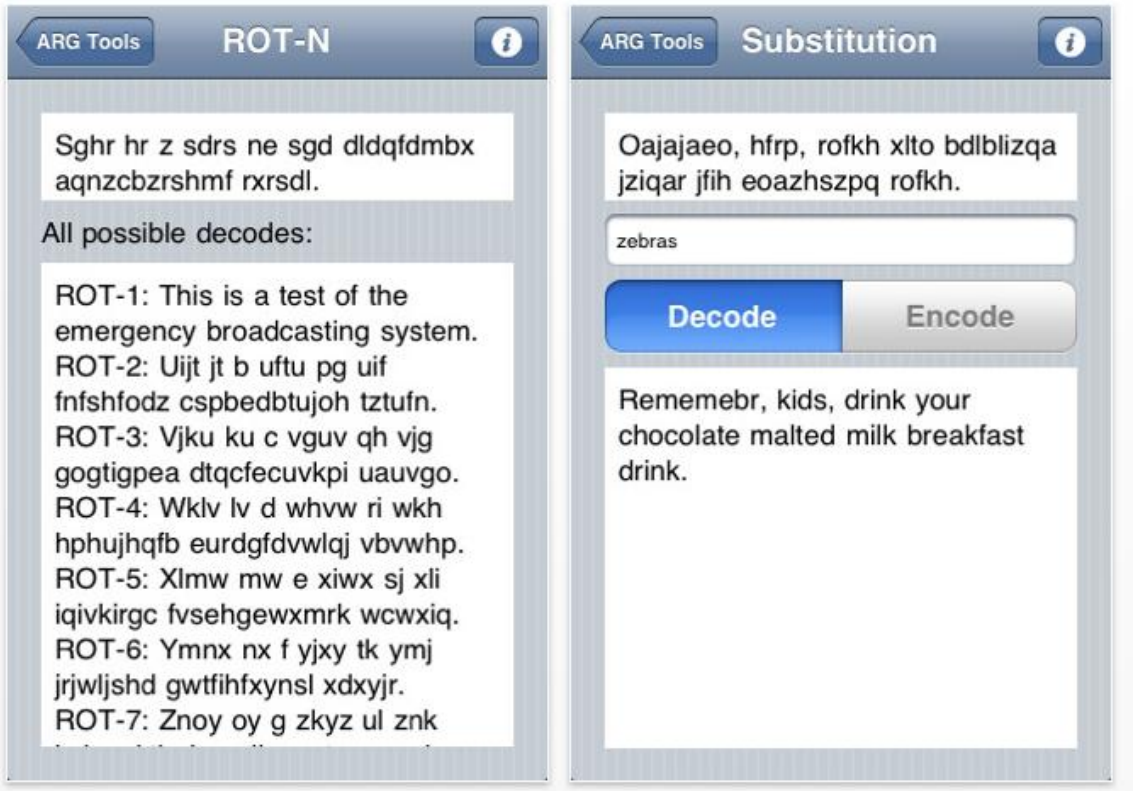


Imagen 7 – Herramientas ARG Tools [12].

En todas estas herramientas se explica el algoritmo que siguen, y la dinámica de la aplicación es simplemente mostrar al usuario cómo funciona el algoritmo descrito mediante un ejemplo guiado. No son juegos, ya que el usuario no resuelve nada por sí mismo, sólo observa cómo se hace.

En la sección de referencias del juego, se incluyen informaciones curiosas como por ejemplo el alfabeto *Braille*, el código *Morse,* los números romanos o el zodiaco chino, entre otros.

Páginas sobre criptografía

En internet existen algunas páginas que ofrecen herramientas interesantes acerca del descifrado de símbolos, como la página de Kriptópilis [18]. En esta página, basada íntegramente en temas relacionados con la criptografía, los usuarios publican textos cifrados, y el resto de usuarios los descifran. Un ejemplo de ello se muestra en la *Imagen 8*.

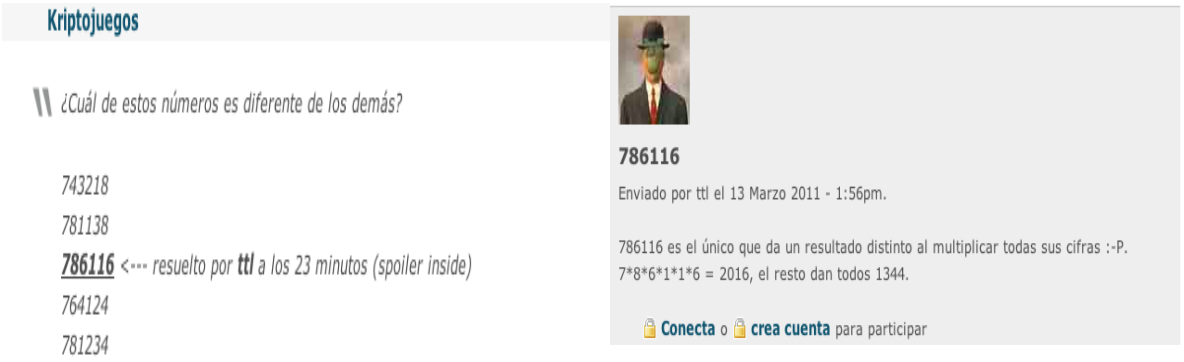


Imagen 8 – Kriptópolis [12].

Se han analizado también páginas como por ejemplo: *Alicia Ramírez* [13], *Great Day Games* [14], el descifrado de criptogramas como en *Gaussianos* [17] o en *CriptoBlog* [18], creación de juegos como en la página de *Titanes DC* [16]. El contenido de estas páginas no se detalla porque su temática, aunque en un principio sí parece muy acorde a la criptografía, después de analizarlos, se ve que la mayoría se basan en averiguar un mensaje cifrado, o en resolver crucigramas.

Después de analizar las opciones más importantes que se ofrecen al usuario interesado en temas relacionados con la criptografía y la seguridad, se ve que aún así hay un gran vacío docente. Muchos de los algoritmos más importantes de criptografía no tienen ningún juego que siga los pasos del algoritmo de manera que al usuario le puedan quedar claros. Tampoco existen juegos que le planteen al usuario un reto, le proporcionen la información acerca del mismo y posteriormente le dejen enfrentarse al él.

2.2. Herramientas Hardware

En este apartado se describirán las herramientas hardware que se han necesitado para el desarrollo del proyecto.

Dispositivo Mac OS X	Dispositivos iOS
Portátil Apple MacBook Pro, procesador 2.2 GHz Intel Core 2 Duo, 2GB de memoria RAM.	• iPad 1 de 16 GB • iPod Touch 3G de 8GB • iPhone 4 de 32 GB

Tabla 1 – Herramientas hardware utilizadas.

Hay que destacar que aunque estas herramientas son las que se han utilizado, bien podrían ser sustituidas por otras similares. El ordenador podría haber sido sustituido por otro de características similares o incluso superiores, pero difícilmente inferiores, ya que a la hora de desarrollar la aplicación el ordenador puede sufrir un descenso notable en su rendimiento.

Debido a la propia naturaleza del proyecto, el ordenador a utilizar debía ser Macintosh, debido a que la plataforma de desarrollo impone esta restricción. No obstante, los dispositivos móviles utilizados indicados en la Tabla 1, podrían haber sido sustituido por:

- iPad 2.
- iPhone 3 y iPhone 3GS.

2.3. Herramientas Software

En este apartado se describirán las herramientas software de las que se cuentan para desarrollar el proyecto.

Tipo	Nombre Software
Sistema operativo	Apple Mac OS X 10.6.7 Snow Leopard
Entorno de programación	Xcode 3.2.5
Entorno de diseño de interfaces	Interface Builder 3.2.5
Herramienta de control de versiones y backup	Dropbox
Tratamiento de imágenes	Photoshop PaintBrush
Planificación	Microsoft Project
Procesador de textos	Microsoft Word
Presentación	Apple Keynote 09

Casos de uso Diagramas de componentes Diagramas de clases Diagramas de secuencia	Altova UModel 2011 Enterprise Edition

Tabla 2 – Herramientas software utilizadas.

3. Gestión del Proyecto

La gestión de proyecto engloba todo lo relacionado con la planificación, presupuesto y metodología empleada en el proyecto.

3.1. Gestión software

En este apartado, se explica la metodología establecida y el ciclo de vida utilizado en el desarrollo del proyecto.

3.1.1. Metodología de Desarrollo

Una metodología de desarrollo de software es un conjunto estándar de conceptos, prácticas y criterios que se utiliza para estructurar, planear y controlar el proceso de desarrollo de sistemas de información. La elección de la metodología hay que hacerla en función de las necesidades específicas de cada proyecto software a desarrollar. Así, hay metodologías que se adaptan muy bien para un tipo de proyecto pero en cambio, para otro, puede resultar nefasto. En el caso de este proyecto, se ha tenido en cuenta que es un proyecto de software pequeño. Un proyecto de software se dice que es pequeño en función de los siguientes criterios[1]:

- Coste del desarrollo del proyecto (si se necesitan menos de dos años hombre para el desarrollo del proyecto).

- Cantidad de personas que se necesitan para desarrollar el software (si se requiere un equipo único de desarrollo de cinco o menos personas).

- Cantidad que se va a producir de software (si la cantidad de código fuente es inferior a las 10000 líneas de código, excluyendo los comentarios).

Para este proyecto, el tiempo de desarrollo es de 8 meses (ver apartado *3.2.1.2. Planificación real*); el coste de personas para desarrollar el software es de 1 integrante; la cantidad de software producido será inferior a las 10000 líneas de código.

El proyecto a desarrollar es un proyecto de tamaño pequeño, por lo tanto, lo propio es la utilización de una metodología ligera.

[1] Los criterios se han obtenido directamente de la descripción que proporciona el estándar de la ESA Lite, en su documento BSSC962 – ES, Capítulo 2, página 7 [19].

Una de las metodologías ligeras adecuadas al tipo de proyecto que nos ocupa, es la metodología ESA [20]. Esta metodología tiene una versión para proyectos pequeños, llamada ESA Lite [19]. En ella se simplifican la documentación, los planes, se combinan diferentes fases y se reduce la formalidad de los requisitos entre otros.

Aún así, en el desarrollo de este proyecto, se ha visto la necesidad de introducir alguna modificación a esta metodología para hacerla apropiada a nuestro proyecto. El resultado de esas modificaciones aplicadas a la metodología ESA Lite pueden comprobarse a lo largo del presente documento.

3.1.2. Ciclo de vida

El ciclo de vida de un proyecto software determina y define las fases por las cuales el proyecto debe pasar.

En el caso de este proyecto, las fases por las que debe pasar se muestran en la siguiente figura:

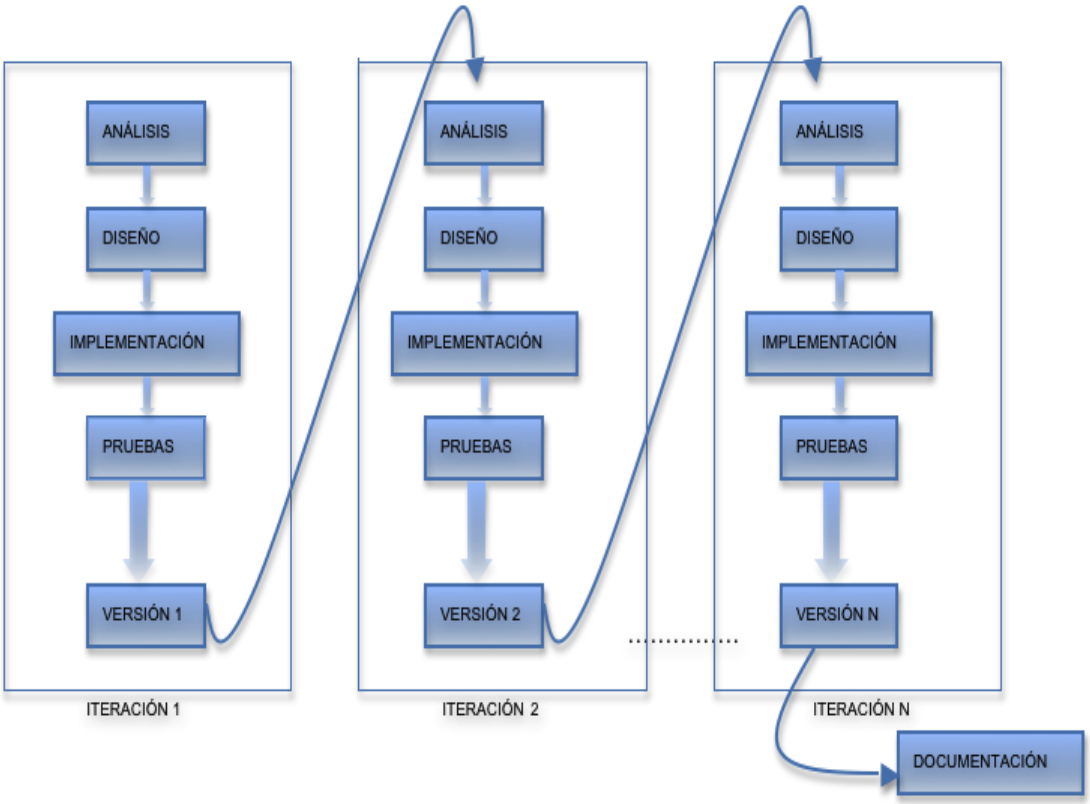


Imagen 9 – Ciclo de vida del software.

La figura anterior (ver *Imagen 9*) muestra las fases por las que pasa el proyecto antes de obtener una versión entregable. Una vez que se entrega la

versión, se debe revisar con el cliente, en este caso el tutor de proyecto, Jorge Blasco. Después de la revisión se obtienen nuevas especificaciones, por lo se debe realizar una nueva iteración para entregar una nueva versión del sistema. Después de las versiones que se consideren oportunas para obtener una versión del sistema final, es cuando se genera la documentación referente al proyecto.

Como se puede ver en la *Imagen 9*, el ciclo de vida que se ha utilizado es un ciclo incremental e iterativo. En este tipo de ciclos de vida se repiten las fases para ir obteniendo diferentes versiones, cada una de ellas más cercana a la versión final deseada.

Cada una de las fases que se han indicado en la *Imagen 9* tiene un cometido dentro del desarrollo del proyecto:

Fase de análisis: Se determinan los objetivos que se persiguen con este proyecto (qué es lo que se espera del proyecto). Esos objetivos se definen a través de los casos de uso con sus correspondientes diagramas de actividad, requisitos de usuario, requisitos software y requisitos hardware.

Fase de diseño: Se determinan los módulos que forman parte del proyecto, el modelo de datos que se necesita, los componentes que conforman la arquitectura del sistema, las clases que implementarán esos componentes y las interfaces con las que contará el sistema. Esta fase define cómo se van a conseguir los objetivos del proyecto.

Fase de implementación: Durante esta fase se implementa la aplicación.

Fase de pruebas: Durante esta fase se verifica el correcto funcionamiento de la aplicación. Esta fase se debe realizar de forma paralela a la fase de implementación. Para cada implementación realizada, se debe comprobar que la funcionalidad obtenida es exactamente igual a la esperada.

Fase de documentación: Fase en la que se procede a la escritura de la presente memoria, donde se detalla todo lo referente al proyecto.

3.2. Integración con otros proyectos

El presente documento hace referencia al proyecto denominado *Minijuegos*, un proyecto que se encuentra integrado en una única aplicación común llamada *Enigmatium*.

Enigmatium es una aplicación para iPhone que está compuesto por un módulo llamado *Minijuegos* (proyecto que se explica a lo largo del presente documento) y un segundo módulo denominado *Aventura.*

Para el desarrollo de *Enigmatium* se ha trabajado de manera conjunta, tanto la autora del proyecto *Aventura*, Judith Medina Gónzalez, como la autora del proyecto *Minijuegos*, Patricia López Peña (autora de este documento). En ambos proyectos el tutor, ha sido Jorge Blasco Alís.

Durante el desarrollo de cada uno de los proyectos, se ha mantenido una comunicación y colaboración activa entre las autoras de *Aventura* y *Minijuegos*, ya que el objetivo principal es el de desarrollar una aplicación común. Para ello ha sido necesario establecer interfaces comunes, el mismo modelo de datos al crear un juego y posteriormente al utilizarlo, el mismo modelo de persistencia de los datos, el mismo método de navegación entre pantallas, etc. Debido a todos los elementos comunes que se comparten, la supervisión por parte de Judith al proyecto de *Minijuegos*, así como la supervisión de Patricia al proyecto de *Aventura*, ha sido fundamental para la realización de la aplicación conjunta. Todas estas tareas de supervisión han sido realizadas y posteriormente aprobadas por el tutor Jorge Blasco.

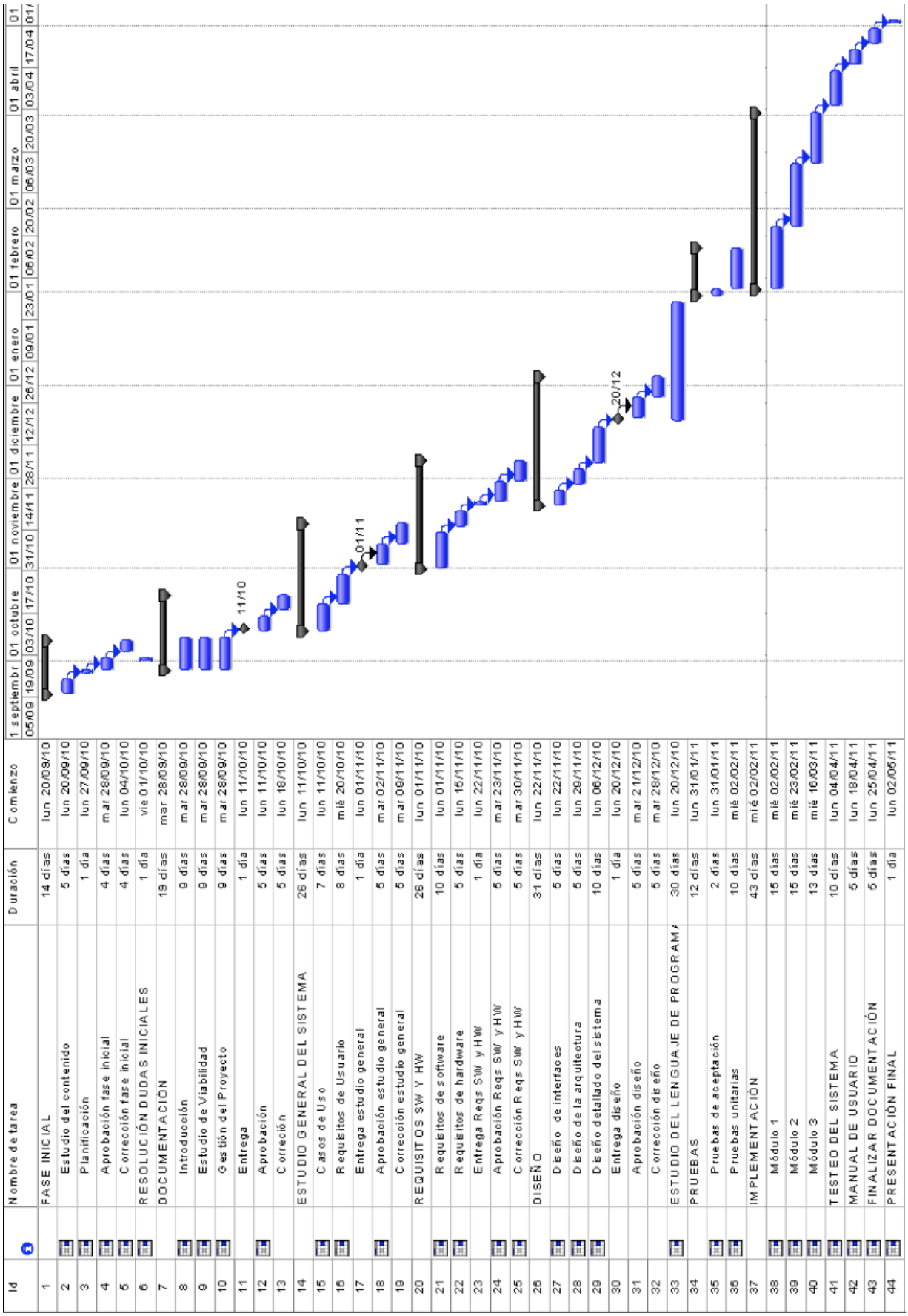
3.2.1. Planificación del proyecto

En esta sección se detallan las tareas más importantes que han sido desarrolladas, así como el esfuerzo en días de cada una de ellas.

Tanto en la planificación inicial como final, las tareas que se incluyen en el diagrama de Gantt son todas aquellas que se han necesitado para el desarrollo de *Enigmatium*, es decir, del proyecto conjunto. No obstante, en cada una de las subsecciones, se detalla cuáles han sido las que conciernen al presente proyecto, *Minijuegos*.

3.2.1.1. Planificación inicial

En esta sección se muestra una planificación inicial mediante un diagrama de Gantt con las tareas realizadas en cada una de sus fases.



Id	Nombre de tarea	Duración	Comienzo
1	FASE INICIAL	14 días	lun 20/09/10
2	Estudio del contenido	5 días	lun 20/09/10
3	Planificación	1 día	lun 27/09/10
4	Aprobación fase inicial	4 días	mar 28/09/10
5	Corrección fase inicial	4 días	lun 04/10/10
6	RESOLUCIÓN DUDAS INICIALES	1 día	vie 01/10/10
7	DOCUMENTACIÓN	19 días	mar 28/09/10
8	Introducción	9 días	mar 28/09/10
9	Estudio de Viabilidad	9 días	mar 28/09/10
10	Gestión del Proyecto	9 días	mar 28/09/10
11	Entrega	1 día	lun 11/10/10
12	Aprobación	5 días	lun 11/10/10
13	Corrección	5 días	lun 18/10/10
14	ESTUDIO GENERAL DEL SISTEMA	26 días	lun 11/10/10
15	Casos de Uso	7 días	lun 11/10/10
16	Requisitos de Usuario	8 días	mié 20/10/10
17	Entrega estudio general	1 día	lun 01/11/10
18	Aprobación estudio general	5 días	mar 02/11/10
19	Corrección estudio general	5 días	mar 09/11/10
20	REQUISITOS SW Y HW	26 días	lun 01/11/10
21	Requisitos de software	10 días	lun 01/11/10
22	Requisitos de hardware	5 días	lun 15/11/10
23	Entrega Reqs SW y HW	1 día	lun 22/11/10
24	Aprobación Reqs SW y HW	5 días	mar 23/11/10
25	Corrección Reqs SW y HW	5 días	mar 30/11/10
26	DISEÑO	31 días	lun 22/11/10
27	Diseño de interfaces	5 días	lun 22/11/10
28	Diseño de la arquitectura	5 días	lun 29/11/10
29	Diseño detallado del sistema	10 días	lun 06/12/10
30	Entrega diseño	1 día	lun 20/12/10
31	Aprobación diseño	5 días	mar 21/12/10
32	Corrección diseño	5 días	mar 28/12/10
33	ESTUDIO DEL LENGUAJE DE PROGRAMA	30 días	lun 20/12/10
34	PRUEBAS	12 días	lun 31/01/11
35	Pruebas de aceptación	2 días	lun 31/01/11
36	Pruebas unitarias	10 días	mié 02/02/11
37	IMPLEMENTACIÓN	43 días	mié 02/02/11
38	Módulo 1	15 días	mié 02/02/11
39	Módulo 2	15 días	mié 23/02/11
40	Módulo 3	13 días	mié 16/03/11
41	TESTEO DEL SISTEMA	10 días	lun 04/04/11
42	MANUAL DE USUARIO	5 días	lun 18/04/11
43	FINALIZAR DOCUMENTACIÓN	5 días	lun 25/04/11
44	PRESENTACIÓN FINAL	1 día	lun 02/05/11

En la planificación inicial se especificaron las fases que se consideraron necesarias para la correcta realización del proyecto.

Fase inicial: El estudio de los contenidos del sistema, planificación del proyecto y aprobación de los objetivos a realizar en el proyecto se estiman desde el 20 de septiembre de 2010 hasta el 7 de octubre de 2010.

Documentación: En esta fase se estima la elaboración de las tres primeras secciones del presente documento, las cuales contienen una introducción, el estudio de viabilidad del sistema y la gestión del proyecto. Para la elaboración de estas tareas se hace una estimación de una semana se duración, desde el 28 de septiembre de 2010 al 8 de octubre de 2010.

Estudio general del sistema: Donde se incluye la elaboración de los casos de uso y los requisitos de usuario concernientes al sistema.

Para los casos de uso se estimó definirlos desde el 11 al 19 de octubre de 2010, mientras que los requisitos de usuario quedarían definidos entre los días 20 y 29 de octubre de 2010.

Requisitos Software y Hardware: Para la definición de los requisitos software se hizo una estimación del 1 de noviembre al 12 de noviembre para que quedaran perfectamente definidos. Para los requisitos hardware se dio una estimación desde el 15 al 19 de noviembre de 2010.

Diseño: Para el diseño de las interfaces se estimó del 22 al 26 de noviembre. Para el diseño de la arquitectura se esperaba trabajar desde el 29 de noviembre al 3 de diciembre de 2010. Finalmente en esta fase de diseño, se planificó incluir el diseño detallado del sistema que se realizaría del 6 al 17 de diciembre de 2010.

Estudio del lenguaje de programación: En esta planificación inicial se contó con el período comprendido desde el 20 de diciembre de 2010 al 28 de enero de 2011 para el estudio del lenguaje de programación y entorno de desarrollo del sistema.

Implementación: La fase de implementación comprendía la implementación de dos módulos en los que inicialmente se iba a dividir la aplicación: parte de producción de juegos y parte de consumición de juegos.

Esta fase al completo de estimó completar en los días comprendidos entre el 31 de enero y el 1 de abril de 2011.

Pruebas: Fase donde se incluirían las pruebas de aceptación del sistema y las pruebas de usabilidad, con una extensión en el tiempo comprendida desde el 21 de marzo al 1 de abril de 2011.

Testeo del sistema: Para probar el correcto funcionamiento del sistema se estimó hacerlo del 4 al 15 de abril de 2011.

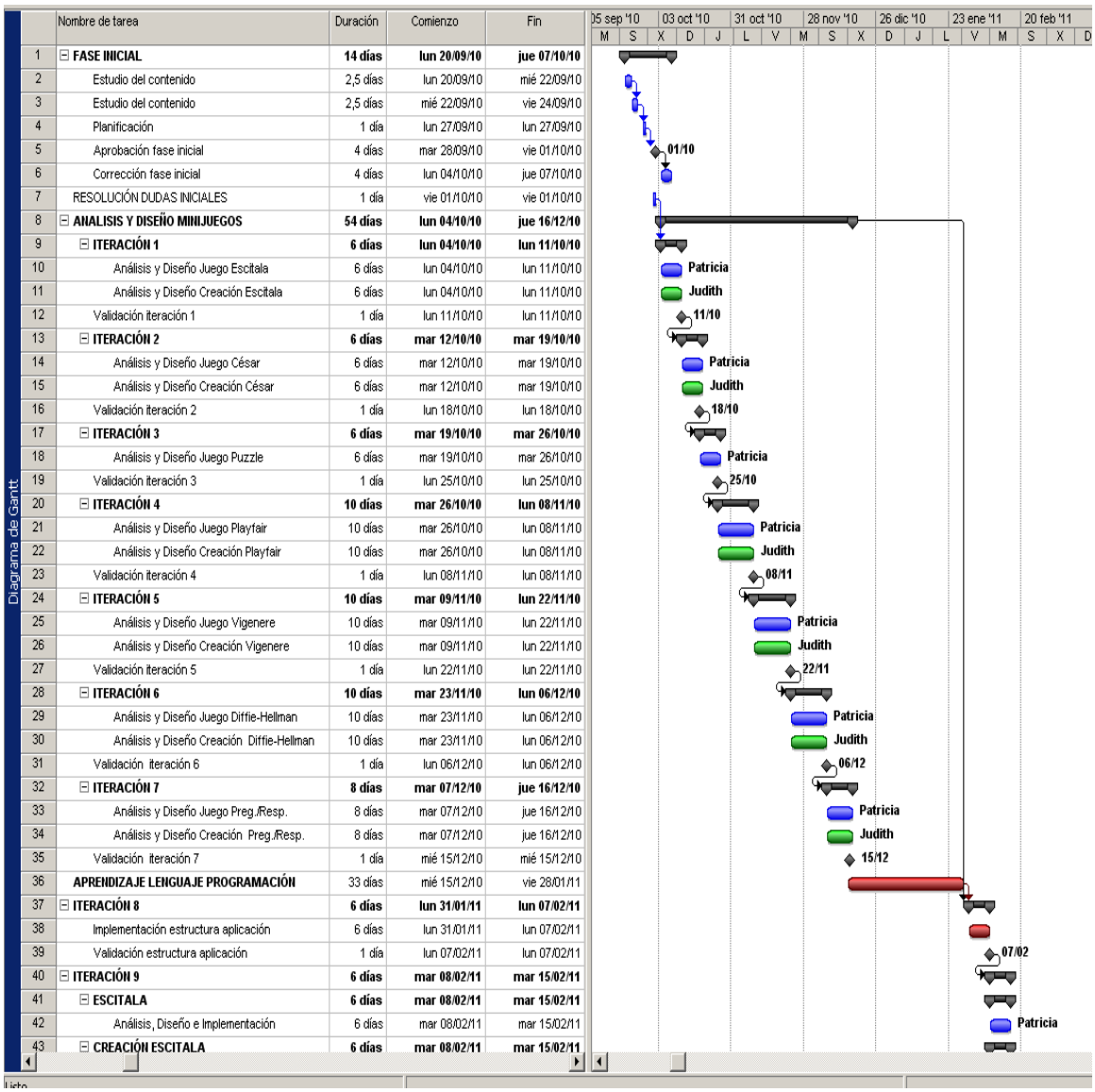
Manual de usuario: Esta fase se realizaría desde el 18 al 22 de abril de 2011.

Finalizar documentación: Para finalizar la documentación restante establecimos la semana del 25 al 29 de abril.

De manera que siguiendo este calendario, con un margen de unos días para imprevistos, se podría haber finalizado a mediados de mayo de 2011.

3.2.1.2. Planificación real

En este apartado se detalla la planificación real del proyecto.



	Nombre de tarea	Duración	Comienzo	Fin
1	**FASE INICIAL**	**14 días**	**lun 20/09/10**	**jue 07/10/10**
2	Estudio del contenido	2,5 días	lun 20/09/10	mié 22/09/10
3	Estudio del contenido	2,5 días	mié 22/09/10	vie 24/09/10
4	Planificación	1 día	lun 27/09/10	lun 27/09/10
5	Aprobación fase inicial	4 días	mar 28/09/10	vie 01/10/10
6	Corrección fase inicial	4 días	lun 04/10/10	jue 07/10/10
7	RESOLUCIÓN DUDAS INICIALES	1 día	vie 01/10/10	vie 01/10/10
8	**ANALISIS Y DISEÑO MINIJUEGOS**	**54 días**	**lun 04/10/10**	**jue 16/12/10**
9	**ITERACIÓN 1**	**6 días**	**lun 04/10/10**	**lun 11/10/10**
10	Análisis y Diseño Juego Escitala	6 días	lun 04/10/10	lun 11/10/10
11	Análisis y Diseño Creación Escitala	6 días	lun 04/10/10	lun 11/10/10
12	Validación iteración 1	1 día	lun 11/10/10	lun 11/10/10
13	**ITERACIÓN 2**	**6 días**	**mar 12/10/10**	**mar 19/10/10**
14	Análisis y Diseño Juego César	6 días	mar 12/10/10	mar 19/10/10
15	Análisis y Diseño Creación César	6 días	mar 12/10/10	mar 19/10/10
16	Validación iteración 2	1 día	lun 18/10/10	lun 18/10/10
17	**ITERACIÓN 3**	**6 días**	**mar 19/10/10**	**mar 26/10/10**
18	Análisis y Diseño Juego Puzzle	6 días	mar 19/10/10	mar 26/10/10
19	Validación iteración 3	1 día	lun 25/10/10	lun 25/10/10
20	**ITERACIÓN 4**	**10 días**	**mar 26/10/10**	**lun 08/11/10**
21	Análisis y Diseño Juego Playfair	10 días	mar 26/10/10	lun 08/11/10
22	Análisis y Diseño Creación Playfair	10 días	mar 26/10/10	lun 08/11/10
23	Validación iteración 4	1 día	lun 08/11/10	lun 08/11/10
24	**ITERACIÓN 5**	**10 días**	**mar 09/11/10**	**lun 22/11/10**
25	Análisis y Diseño Juego Vigenere	10 días	mar 09/11/10	lun 22/11/10
26	Análisis y Diseño Creación Vigenere	10 días	mar 09/11/10	lun 22/11/10
27	Validación iteración 5	1 día	lun 22/11/10	lun 22/11/10
28	**ITERACIÓN 6**	**10 días**	**mar 23/11/10**	**lun 06/12/10**
29	Análisis y Diseño Juego Diffie-Hellman	10 días	mar 23/11/10	lun 06/12/10
30	Análisis y Diseño Creación Diffie-Hellman	10 días	mar 23/11/10	lun 06/12/10
31	Validación iteración 6	1 día	lun 06/12/10	lun 06/12/10
32	**ITERACIÓN 7**	**8 días**	**mar 07/12/10**	**jue 16/12/10**
33	Análisis y Diseño Juego Preg./Resp.	8 días	mar 07/12/10	jue 16/12/10
34	Análisis y Diseño Creación Preg./Resp.	8 días	mar 07/12/10	jue 16/12/10
35	Validación iteración 7	1 día	mié 15/12/10	mié 15/12/10
36	**APRENDIZAJE LENGUAJE PROGRAMACIÓN**	**33 días**	**mié 15/12/10**	**vie 28/01/11**
37	**ITERACIÓN 8**	**6 días**	**lun 31/01/11**	**lun 07/02/11**
38	Implementación estructura aplicación	6 días	lun 31/01/11	lun 07/02/11
39	Validación estructura aplicación	1 día	lun 07/02/11	lun 07/02/11
40	**ITERACIÓN 9**	**6 días**	**mar 08/02/11**	**mar 15/02/11**
41	**ESCITALA**	**6 días**	**mar 08/02/11**	**mar 15/02/11**
42	Análisis, Diseño e Implementación	6 días	mar 08/02/11	mar 15/02/11
43	**CREACIÓN ESCITALA**	**6 días**	**mar 08/02/11**	**mar 15/02/11**

Imagen 10 – Planificación Real. Parte 1.

Como se puede observar en la imagen de la planificación real (*ver Imagen 10*), las fases del proyecto han sido modificadas considerablemente con respecto a la planificación inicial.

El cometido de cada una de las fases es el siguiente:

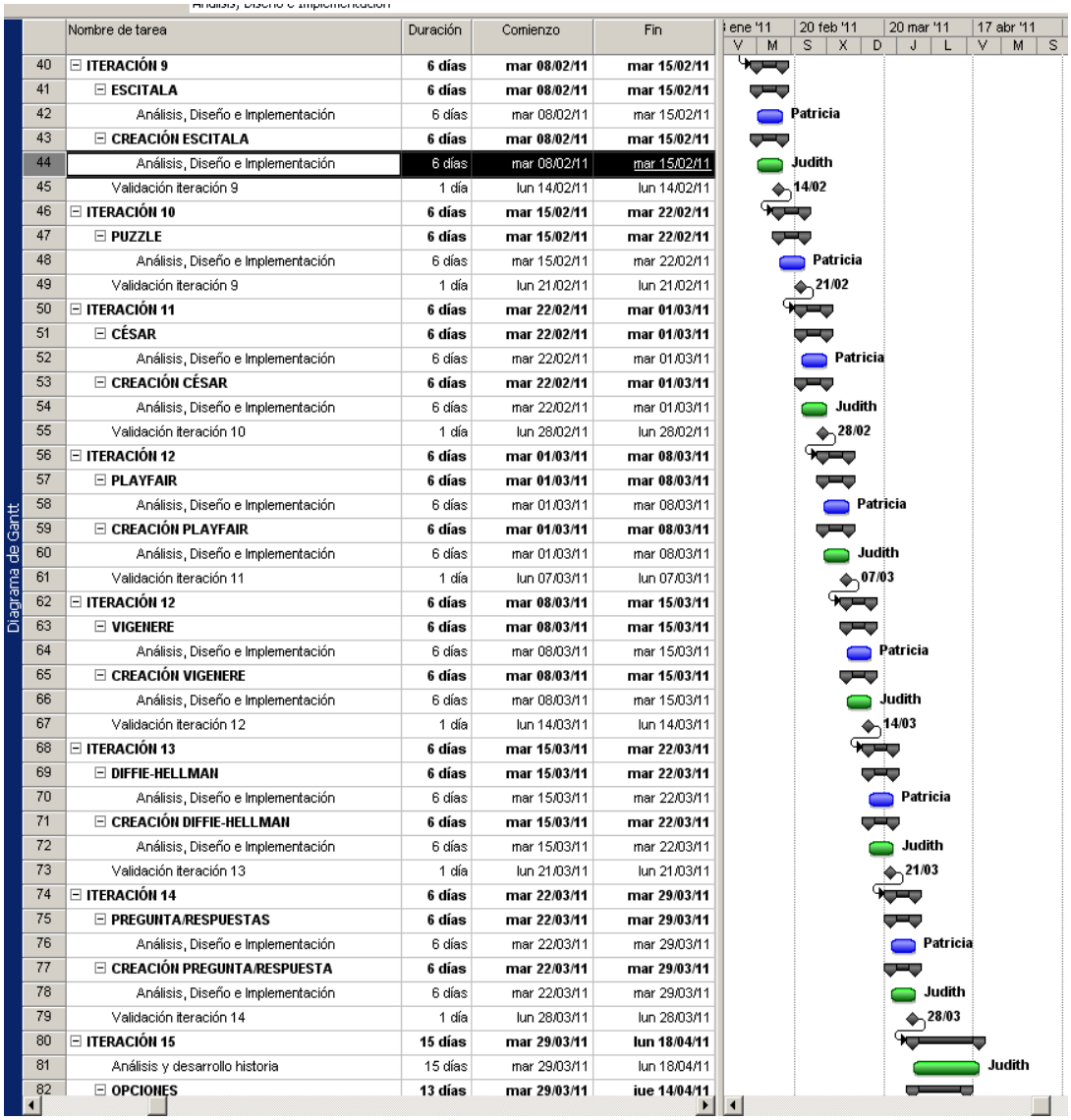
Fase inicial: El estudio de los contenidos del sistema, planificación del proyecto y aprobación de los objetivos a realizar en el proyecto comienzan el 20 de septiembre de 2010, finalizando el 7 de octubre de 2010.

Análisis y diseño de minijuegos: Esta fase está compuesta por 7 iteraciones. Cada una de ellas contiene el análisis y diseño de cada uno de los juegos, así como su validación por parte del tutor. Dentro del diseño de cada juego se incluye tanto la lógica del mismo, como sus prototipos de alto nivel. Comenzando en la iteración 1 hasta la iteración 7, los juegos tratados son *Escitala*, *César*, *Puzle*, *Playfair*, *Vigenere*, *Diffie-Hellman* y *Pregunta*/*Respuesta*.

Esta fase se extiende desde el 4 de octubre de 2010 hasta el 8 de diciembre de ese mismo año. Durante esta fase, la dedicación diaria en horas por parte de la autora se ve reducida a media jornada, debido a temas laborales y académicos.

Aprendizaje del lenguaje de programación: Esta fase se extiende desde el 15 de diciembre de 2010 al 28 de enero de 2011. Durante este mes se procede al estudio y documentación del lenguaje y entorno de programación (Objective C y entorno de programación SDK de iPhone). Cabe destacar que durante el mes de documentación y aprendizaje, la autora también se dedica al estudio de otras materias pertenecientes a la carrera, por lo que no se puede dedicar plenamente el tiempo a la formación referente a la aplicación.

Iteración 8: El 31 de enero de 2011 comienza la implementación real de la estructura general de la aplicación, finalizándola el 7 de febrero, con la correspondiente validación por parte del tutor. Esta iteración se realiza de manera conjunta con Judith, autora de *Aventura*.



	Nombre de tarea	Duración	Comienzo	Fin
40	**ITERACIÓN 9**	**6 días**	**mar 08/02/11**	**mar 15/02/11**
41	**ESCITALA**	**6 días**	**mar 08/02/11**	**mar 15/02/11**
42	Análisis, Diseño e Implementación	6 días	mar 08/02/11	mar 15/02/11
43	**CREACIÓN ESCITALA**	**6 días**	**mar 08/02/11**	**mar 15/02/11**
44	Análisis, Diseño e Implementación	6 días	mar 08/02/11	mar 15/02/11
45	Validación iteración 9	1 día	lun 14/02/11	lun 14/02/11
46	**ITERACIÓN 10**	**6 días**	**mar 15/02/11**	**mar 22/02/11**
47	**PUZZLE**	**6 días**	**mar 15/02/11**	**mar 22/02/11**
48	Análisis, Diseño e Implementación	6 días	mar 15/02/11	mar 22/02/11
49	Validación iteración 9	1 día	lun 21/02/11	lun 21/02/11
50	**ITERACIÓN 11**	**6 días**	**mar 22/02/11**	**mar 01/03/11**
51	**CÉSAR**	**6 días**	**mar 22/02/11**	**mar 01/03/11**
52	Análisis, Diseño e Implementación	6 días	mar 22/02/11	mar 01/03/11
53	**CREACIÓN CÉSAR**	**6 días**	**mar 22/02/11**	**mar 01/03/11**
54	Análisis, Diseño e Implementación	6 días	mar 22/02/11	mar 01/03/11
55	Validación iteración 10	1 día	lun 28/02/11	lun 28/02/11
56	**ITERACIÓN 12**	**6 días**	**mar 01/03/11**	**mar 08/03/11**
57	**PLAYFAIR**	**6 días**	**mar 01/03/11**	**mar 08/03/11**
58	Análisis, Diseño e Implementación	6 días	mar 01/03/11	mar 08/03/11
59	**CREACIÓN PLAYFAIR**	**6 días**	**mar 01/03/11**	**mar 08/03/11**
60	Análisis, Diseño e Implementación	6 días	mar 01/03/11	mar 08/03/11
61	Validación iteración 11	1 día	lun 07/03/11	lun 07/03/11
62	**ITERACIÓN 12**	**6 días**	**mar 08/03/11**	**mar 15/03/11**
63	**VIGENERE**	**6 días**	**mar 08/03/11**	**mar 15/03/11**
64	Análisis, Diseño e Implementación	6 días	mar 08/03/11	mar 15/03/11
65	**CREACIÓN VIGENERE**	**6 días**	**mar 08/03/11**	**mar 15/03/11**
66	Análisis, Diseño e Implementación	6 días	mar 08/03/11	mar 15/03/11
67	Validación iteración 12	1 día	lun 14/03/11	lun 14/03/11
68	**ITERACIÓN 13**	**6 días**	**mar 15/03/11**	**mar 22/03/11**
69	**DIFFIE-HELLMAN**	**6 días**	**mar 15/03/11**	**mar 22/03/11**
70	Análisis, Diseño e Implementación	6 días	mar 15/03/11	mar 22/03/11
71	**CREACIÓN DIFFIE-HELLMAN**	**6 días**	**mar 15/03/11**	**mar 22/03/11**
72	Análisis, Diseño e Implementación	6 días	mar 15/03/11	mar 22/03/11
73	Validación iteración 13	1 día	lun 21/03/11	lun 21/03/11
74	**ITERACIÓN 14**	**6 días**	**mar 22/03/11**	**mar 29/03/11**
75	**PREGUNTA/RESPUESTAS**	**6 días**	**mar 22/03/11**	**mar 29/03/11**
76	Análisis, Diseño e Implementación	6 días	mar 22/03/11	mar 29/03/11
77	**CREACIÓN PREGUNTA/RESPUESTA**	**6 días**	**mar 22/03/11**	**mar 29/03/11**
78	Análisis, Diseño e Implementación	6 días	mar 22/03/11	mar 29/03/11
79	Validación iteración 14	1 día	lun 28/03/11	lun 28/03/11
80	**ITERACIÓN 15**	**15 días**	**mar 29/03/11**	**lun 18/04/11**
81	Análisis y desarrollo historia	15 días	mar 29/03/11	lun 18/04/11
82	**OPCIONES**	**13 días**	**mar 29/03/11**	**iue 14/04/11**

Imagen 11 – Planificación Real. Parte 2.

Iteración 9: Se lleva a cabo el análisis, diseño, implementación (con sus correspondientes pruebas) y validación por parte del tutor del juego de *Escítala.* Esta fase se realiza entre el día 7 de febrero de 2011 el 15 de febrero de 2011.

Iteración 10: Se lleva a cabo el análisis, diseño, implementación (con sus correspondientes pruebas) y validación por parte del tutor del juego de *Puzle.* Esta fase se realiza entre el día 15 de febrero de 2011 y el 22 de febrero de 2011.

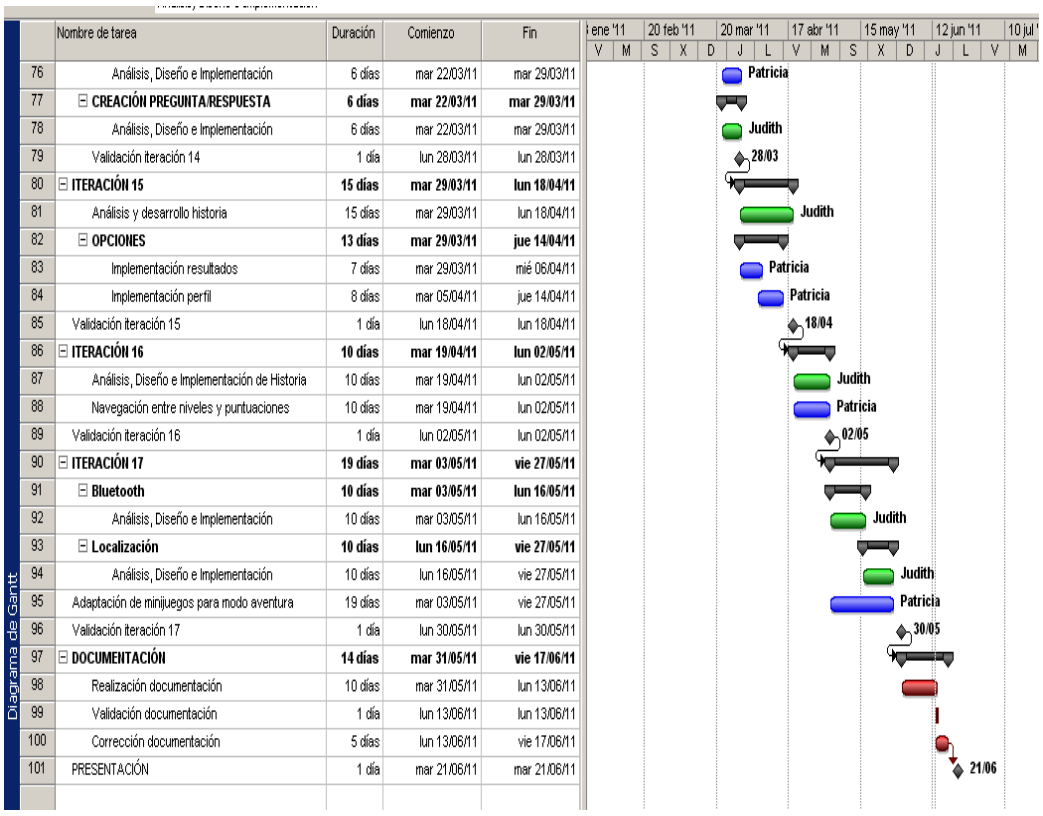
Iteración 11: Se lleva a cabo el análisis, diseño, implementación (con sus correspondientes pruebas) y validación por parte del tutor del juego de *César.* Esta fase se realiza entre el día 22 de febrero de 2011 y el 1 de marzo de 2011.

Iteración 12: Se lleva a cabo el análisis, diseño, implementación (con sus correspondientes pruebas) y validación por parte del tutor del juego de *Playfair.* Esta fase se realiza entre el día 1 de marzo de 2011 y el 8 de marzo de 2011.

Iteración 12: Se lleva a cabo el análisis, diseño, implementación (con sus correspondientes pruebas) y validación por parte del tutor del juego de *Vigenere.* Esta fase se realiza entre el día 8 de marzo de 2011 y el 15 de marzo de 2011.

Iteración 13: Se lleva a cabo el análisis, diseño, implementación (con sus correspondientes pruebas) y validación por parte del tutor del juego de *Diffie-Hellman.* Esta fase se realiza entre el día 15 de marzo de 2011 y el 22 de marzo de 2011.

Iteración 14: Se lleva a cabo el análisis, diseño, implementación (con sus correspondientes pruebas) y validación por parte del tutor del juego de *Pregunta/Respuesta.* Esta fase se realiza entre el día 22 de marzo de 2011 y el 29 de marzo de 2011.



	Nombre de tarea	Duración	Comienzo	Fin
76	Análisis, Diseño e Implementación	6 días	mar 22/03/11	mar 29/03/11
77	**CREACIÓN PREGUNTA/RESPUESTA**	**6 días**	**mar 22/03/11**	**mar 29/03/11**
78	Análisis, Diseño e Implementación	6 días	mar 22/03/11	mar 29/03/11
79	Validación iteración 14	1 día	lun 28/03/11	lun 28/03/11
80	**ITERACIÓN 15**	**15 días**	**mar 29/03/11**	**lun 18/04/11**
81	Análisis y desarrollo historia	15 días	mar 29/03/11	lun 18/04/11
82	**OPCIONES**	**13 días**	**mar 29/03/11**	**jue 14/04/11**
83	Implementación resultados	7 días	mar 29/03/11	mié 06/04/11
84	Implementación perfil	8 días	mar 05/04/11	jue 14/04/11
85	Validación iteración 15	1 día	lun 18/04/11	lun 18/04/11
86	**ITERACIÓN 16**	**10 días**	**mar 19/04/11**	**lun 02/05/11**
87	Análisis, Diseño e Implementación de Historia	10 días	mar 19/04/11	lun 02/05/11
88	Navegación entre niveles y puntuaciones	10 días	mar 19/04/11	lun 02/05/11
89	Validación iteración 16	1 día	lun 02/05/11	lun 02/05/11
90	**ITERACIÓN 17**	**19 días**	**mar 03/05/11**	**vie 27/05/11**
91	**Bluetooth**	**10 días**	**mar 03/05/11**	**lun 16/05/11**
92	Análisis, Diseño e Implementación	10 días	mar 03/05/11	lun 16/05/11
93	**Localización**	**10 días**	**lun 16/05/11**	**vie 27/05/11**
94	Análisis, Diseño e Implementación	10 días	lun 16/05/11	vie 27/05/11
95	Adaptación de minijuegos para modo aventura	19 días	mar 03/05/11	vie 27/05/11
96	Validación iteración 17	1 día	lun 30/05/11	lun 30/05/11
97	**DOCUMENTACIÓN**	**14 días**	**mar 31/05/11**	**vie 17/06/11**
98	Realización documentación	10 días	mar 31/05/11	lun 13/06/11
99	Validación documentación	1 día	lun 13/06/11	lun 13/06/11
100	Corrección documentación	5 días	lun 13/06/11	vie 17/06/11
101	PRESENTACIÓN	1 día	mar 21/06/11	mar 21/06/11

Imagen 12 – Planificación Real. Parte 3.

Iteración 15: Se lleva a cabo la implementación de las funcionalidades relativas a las puntuaciones y al perfil. Esta fase se realiza entre el día 29 de marzo de 2011 y el 14 de abril de 2011.

Iteración 16: Se lleva a cabo la navegación entre los diferentes niveles de los juegos así como la gestión de las puntuaciones. Esta fase se realiza entre el día 19 de abril de 2011 y el 2 de mayo de 2011.

Iteración 17: En esta iteración se procede a la adaptación de los minijuegos implementados en el proyecto *Minijuegos* para su correcto funcionamiento dentro del proyecto de *Aventura.* Esta fase se realiza entre el día 3 de mayo de 2011 y el 27 de mayo de 2011.

Documentación: Elaboración del presente documento que comienza el 30 de mayo y finaliza el 17 de junio.

Dando unos días de margen, la presentación se realizará el día 21 de junio.

3.2.1.3. Comentarios acerca de las planificaciones

Como se puede observar comparando las dos planificaciones aunque la fecha de finalización del proyecto sólo difiere en un mes, las tareas que se han realizado a lo largo de los 9 meses han sido muy diferentes las reales de las esperadas.

Mientras que en la planificación inicial se le dio mucha más importancia a la documentación y mucho menos a la implementación, en realidad la implementación ha ocupado más del 50% del tiempo total del proyecto.

En la planificación inicial no se contó con que los autores del proyecto tuviesen reducciones de las horas estimadas debido a temas laborales. Fue un imprevisto que surgió según pasaron los meses.

En un primer momento se pensó utilizar una metodología clásica en cascada, pero a medida que se empezó a desarrollar el proyecto, se hizo necesario cambiar el tipo de metodología y seguir una metodología incremental iterativa. Observando tanto la planificación inicial como la planificación final, se pueden apreciar estas diferencias.

Cabe destacar que el desarrollo de ambos módulos ha sido realizado en un espacio de tiempo equitativo, ya que se ha trabajado de manera paralela y en continua comunicación.

3.3. Análisis de Costes

En esta sección se realiza un análisis de costes con el fin de valorar económicamente la aplicación desarrollada. Los costes que se van a analizar son los derivados de los recursos humanos o personal, los derivados del equipamiento (entre los que se incluyen el hardware y el software) y los costes indirectos.

Se incluirá un análisis de costes inicial, basado en la planificación inicial (ver el apartado *3.2.1.1 Planificación inicial,* de este mismo documento), así como un análisis de costes final, basado a su vez en la planificación final (ver el apartado *3.2.1.2 Planificación real* de este mismo documento). Estos dos análisis muestran la desviación en la que se ha incurrido.

3.3.1. Análisis de costes inicial estimado

3.3.1.1. Estimación del coste de personal

En el coste de personal no se incluye una diferenciación entre los diferentes roles que pueden formar parte del proyecto (gestor de proyecto, analista/diseñador, diseñador de interfaces de usuario o programador), ya que todos estos roles fueron realizados por la misma persona.

Cargo	€/Hora	Núm. Horas	Total
Ingeniero en Informática	20	1.192 horas	**23.840 €**

Tabla 3 – Estimación coste del personal.

El cálculo se ha realizado en base a las tablas de cotización de la Seguridad Social [22]. Estos salarios son en bruto y no incluyen ni IRPF ni cotización a la Seguridad Social.

El cálculo del salario percibido de un Ingeniero Informático se basa en las tablas de cotización de la Seguridad Social donde la base mínima de cotización es de 1.045€ y la base máxima es de 3.230€ (para el año 2011). Si se tiene en cuenta una jornada laboral de 8 horas al día con 20 días laborales al mes, percibiendo el máximo de cotización, se obtiene un salario de 20€/hora.

3.3.1.2. Estimación del coste de hardware

Dispositivo	Coste Unitario	Tiempo vida útil	Tiempo de uso	Coste para el proyecto
Portátil Apple MacBook Pro	1.749€	48 meses	8 meses	291,5€
iPad de 1ª generación de 16 GB	479€	48 meses	8 meses	79,83€
iPod Touch 3G de 8 GB	229€	48 meses	8 meses	38,16€
iPhone 4 de 32 GB	699€	48 meses	8 meses	116,5€
TOTAL	**3.385€**	48 meses	8 meses	**525,99€**

Tabla 4 – Estimación coste de hardware.

El presupuesto se ha realizado siguiendo los precios que figuran en la página oficial de Apple, en Apple Store [24].

3.3.1.3. Estimación del coste de software

Software	Coste Licencia/Año
Mac Box Set (Apple Mac OS 10.6.7 Snow Leopard + iWork '09)	129€
XCode, Interface Builder	0€
Dropbox	0€
Photoshop	478€
Paintbrush	0€
Microsoft Office para Mac	149,99€
Microsoft Project	1.300€
iPhone Developer Program	79,82€

Altova UModel 2011 Enterprise Edition	0€ (Versión prueba 30 días)
TOTAL	**2.136,81€**

Tabla 5 – Estimación coste del software.

Los costes anteriormente indicados figuran en las páginas oficiales de cada uno de los programas utilizados [23, 24, 25 y 26].

3.3.1.4. Estimación de costes indirectos

Concepto	Precio Mensual	Tiempo de uso	Total
Conexión a Internet	39€	8 meses	312€
Luz	50€	8 meses	400€
Total	89€	8 meses	**712€**

Tabla 6 – Estimación costes indirectos.

3.3.1.5. Estimación de costes totales

Concepto	Coste
Coste de personal	23.840€
Coste de hardware	525,99€
Coste de software	2.136,81€
Costes indirectos	712€
TOTAL	**27.214,8€**

Tabla 7 – Estimación costes totales.

Como se puede observar en la *tabla 7*, la estimación sobre el coste total del proyecto asciende a **27.214,8€.** En el siguiente apartado se analizan los costes reales y la diferencia entre el coste estimado y el real.

3.3.2. Análisis de costes reales

En el coste de los recursos humanos, igual que se ha hecho en el análisis de costes iniciales estimados (ver apartado anterior, 3.3.1 *Análisis de costes inicial estimado*) no se incluye una diferenciación entre los diferentes roles que pueden formar parte del proyecto (gestor de proyecto, analista/diseñador, diseñador de interfaces de usuario o programador), ya que todos estos roles fueron realizados por la misma persona.

En cambio, sí hay una diferenciación entre el ingeniero en informática y el probador, ya que dentro de éste último se debe incluir a todas las personas que colaboraron activamente con la realización de las pruebas.

3.3.2.1. Coste real de personal

Cargo	€/Horas	Núm. Horas	Total
Ingeniero en Informática	20	1132	22.640 €
Probador	10	12	120 €
TOTAL			**22.760 €**

Tabla 8 – Coste real del personal.

El cálculo se ha realizado en base a las tablas de cotización de la Seguridad Social [22]. Estos salarios son en bruto y no incluyen ni IRPF ni cotización a la Seguridad Social.

3.3.2.2. Coste real de hardware

Dispositivo	Coste Unitario	Tiempo vida útil	Tiempo de uso	Coste para el proyecto
Portátil Apple MacBook Pro	1.749€	48 meses	9 meses	327,94€
iPad de 1ª generación de 16 GB	479€	48 meses	9 meses	89,81€
iPod Touch 3G de 8 GB	229€	48 meses	9 meses	42,94€
iPhone 4 de 32 GB	699€	48 meses	9 meses	131,06€

TOTAL	**3.385€**	48 meses	9 meses	**591,75€**

Tabla 9 – Coste real del hardware.

3.3.2.3. Coste real de software

Software	Coste Licencia/Año
Mac Box Set (Apple Mac OS 10.6.7 Snow Leopard + iWork '09)	129€
XCode, Interface Builder	0€
Dropbox	0€
Photoshop	478€
Paintbrush	0€
Microsoft Office para Mac	149,99€
Microsoft Project	1.300€
iPhone Developer Program	79,82€
Altova UModel 2011 Enterprise Edition	0€ (Versión prueba 30 días)
TOTAL	**2.136,81€**

Tabla 10 – Coste real de software.

Los costes indicados en la *Tabla 10*, figuran en las páginas oficiales de cada uno de los programas utilizados [23, 24, 25 y 26].

3.3.2.4. Costes indirectos reales

Concepto	Precio Mensual	Tiempo de uso	Total
Conexión a Internet	39€	9 meses	351€
Luz	50€	9 meses	450€

Total	89€	9 meses	**801€**

Tabla 11 – Coste indirecto real.

3.3.2.5. Costes totales reales

Concepto	Estimación total	Coste total real	Diferencia
Coste de personal	23.840€	22.760€	1080 (-)
Coste de hardware	525,99€	591,75€	65,76(+)
Coste de software	2.136,81€	2.136,81€	-
Costes indirectos	712€	801€	89 (+)
TOTAL	**27.214,8€**	**26.289,56€**	**925,24 (-)**

Tabla 12 – Coste total real.

El coste real del proyecto ha sido **26.289,56€**, con una desviación de **925,24€** con respecto a la estimación inicial.

La diferencia de horas trabajadas entre la estimación y lo real radican en que al hacer la estimación se decidió hacer jornadas de 8 horas de lunes a viernes. Debido a temas laborales, esos períodos de 8 horas tuvieron que reducirse hasta las 4 horas diarias, incluyendo fines de semana. Pese al descenso en las horas dedicadas al proyecto, debido a la incorporación de los fines de semana, el proyecto se ha podido concluir un mes después de lo esperado con una reducción en el número de horas dedicadas al proyecto.

3.4. Estudio de Mercado para la Venta de la Aplicación

Un objetivo secundario de la presente aplicación es obtener un beneficio económico. Al ser una aplicación creada por la unión de dos proyectos hay que tener en cuenta los gastos de ambos y realizar un estudio de los posibles beneficios futuros que se pudiesen producir en caso de la venta de la aplicación para ambas desarrolladoras. Para el análisis, se va a considerar la venta de la aplicación a través de la tienda de Apple, conocida como *Apple Store*. En este análisis no se considera la venta a un cliente para su explotación.

Para poder publicar una aplicación en App Store, es necesario obtener una Licencia como desarrollador de iOS, “iPhone Developer Program”, que

cuesta 79.82€ por año. El beneficio que obtiene el desarrollador es del 70% y Apple se queda con un 30% de los beneficios totales.

La aplicación puede generar beneficios de dos formas diferentes:

- Venta ordinaria: Poner un precio a la aplicación por cada descarga producida.

- Utilizar el sistema publicitario de Apple iAd, de tal forma que permita a los usuarios descargarse la aplicación de forma gratuita y recibir beneficios por las impresiones de la publicidad o los click encima de la publicidad introducida.

Se realizó un estudio para determinar la mejor tienda de aplicaciones desde el punto de vista de los desarrolladores, según el estudio se extrae que las aplicaciones de pago que podemos encontrar en el App Store tienen un mayor índice de éxito que aquellas que se encuentran en otras plataformas como Android. La comparativa entre las aplicaciones se puede observar en la *Imagen 13*.

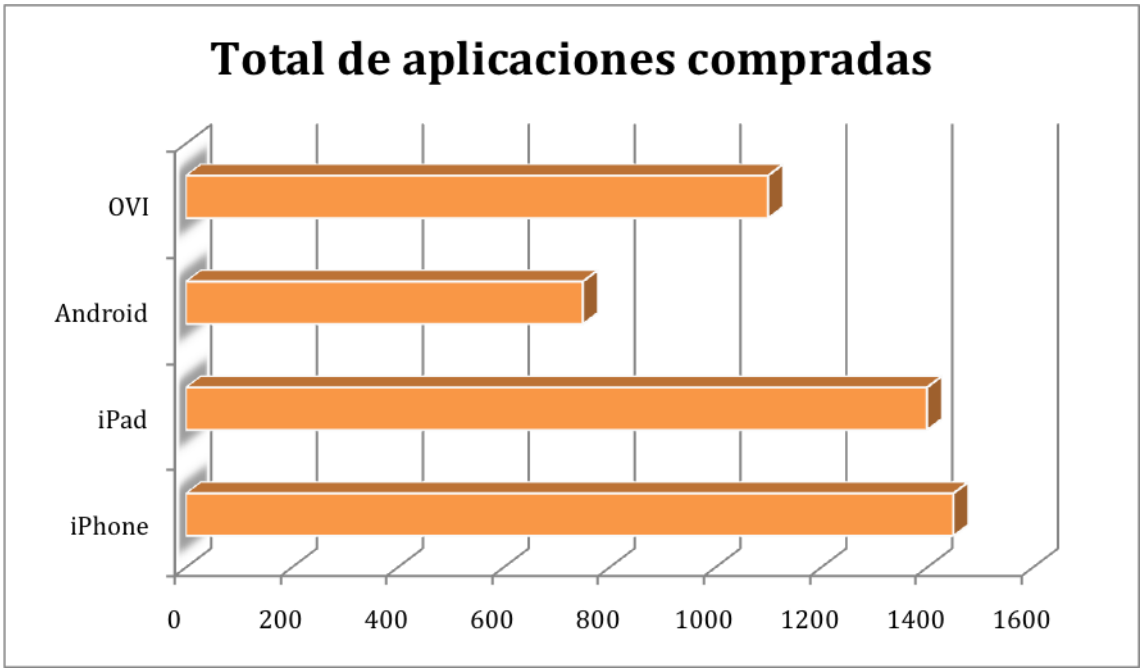


Imagen 13 – Total aplicaciones compradas.

Como ya se comento anteriormente la aplicación es el conjunto de dos proyectos, cuyos presupuestos finales son:

- *ENIGMATIUM-MINIJUEGOS: 26.289,56€.*

- *ENIGMATIUM-AVENTURA: 29.829,56€.*

El presupuesto total de la aplicación es de 56.116,16€. Por tanto se deben obtener los gastos mínimos más un 20% de beneficio que se ha estableci-

do obtener entre las desarrolladoras. Por tanto el importe final será de 67.339,39€, siendo 11.223, 23€ el beneficio obtenido de la aplicación y finalmente será divido entre ambos proyectos.

3.4.1.1. Venta Ordinaria no gratuita

Como ya se ha comentado el importe económico final deseado es de 67.339,39€. El gráfico que se muestra en la *imagen 14* muestra el número de aplicaciones que se deberían vender para obtener el valor económico deseado, para ello se muestran diferentes valores de venta de la aplicación. Hay que tener en cuenta que Apple se queda con el 30% de los beneficios.

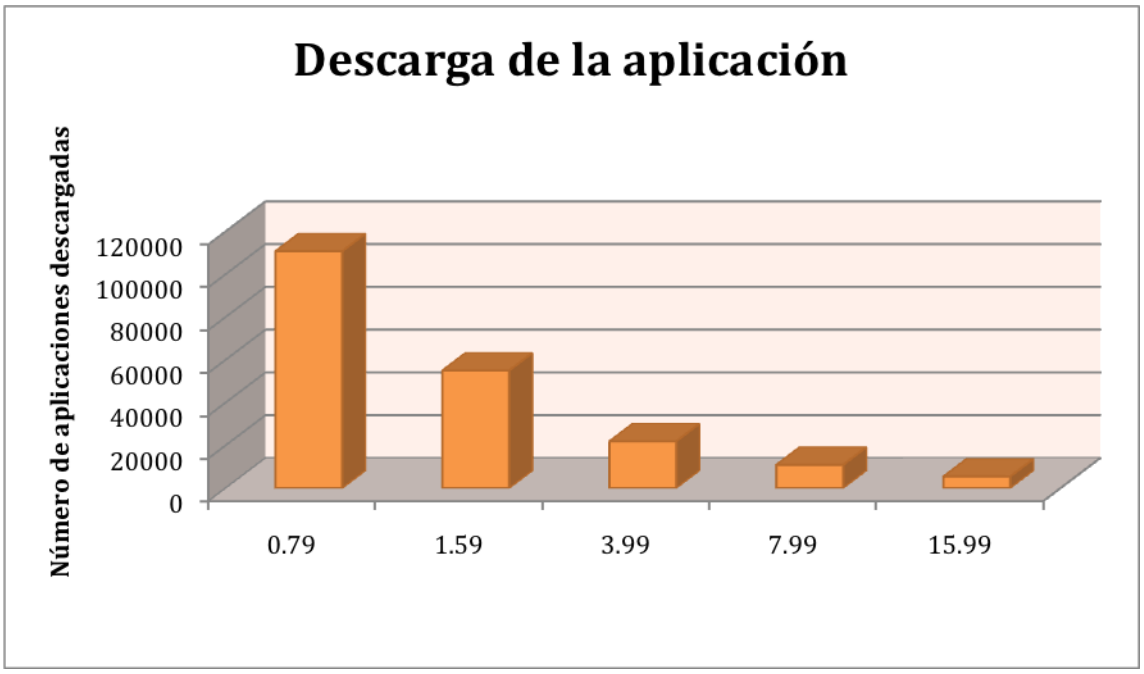


Imagen 14 – Descarga de la aplicación.

En la *Tabla 13* se muestra el tiempo estimado necesario para alcanzar el importe económico establecido. Se ha estimado 30 descargas diarias como media. Si se observa la tabla, se concluye que para sacar el beneficio estimado en un periodo de tiempo aceptable, se tendría que vender la aplicación a 3,99€ y esperar una media de 2 años o venderla a 7,99€ y en un año se alcanzaría el objetivo marcado. Pero hay que ser realistas y aunque como se comento en la introducción el sistema de Apple, tiene mayor acogida a las compras. Una aplicación con valor de 7,99€ tendrá una descarga diaria menor que una aplicación cuyo coste sea menor. Por tanto se va a escoger la venta del producto a 3,99€.

Precio	Número de unidades	Descargas diarias	Años estimados
0,79€	110811	30	9,85
1,59€	55.057	30	4,89
3,99€	21.940	30	1,95
7,99€	10.956	30	0,97
15,99 €	5.474	30	0,49

Tabla 13 – Tiempo para obtener beneficios con número descargas fijo.

El tiempo establecido de 2 años es orientativo, pero pueden darse desviaciones importante según las descargas diarias de la aplicación, por tanto se va a realizar a continuación otro estudio con el precio de 3,99€ pero en este caso las descargas diarias serán distintas para observar las diferentes variaciones.

Precio	Número de unidades	Descargas diarias	Años estimados
3,99	11.0811,65	5	11,70
3,99	55.057,36	10	5,85
3,99	21.940,15	15	3,90
3,99	10.956,35	20	2,93
3,99	5.474,75	25	2,34
3,99	5.474,75	30	1,95

3,99	5.474,75	35	1,67
3,99	5.474,75	40	1,46

Tabla 14 – Tiempo para obtener beneficios con precio fijo.

Como se puede observar en la *Tabla 14*, si el número de descargas diarias es menor de 10, la recuperación de beneficios se hará muy larga, por el contrario si ronda los 25 a 35 descargas diarias la recuperación del beneficio será sostenida y posible. El número de descargas es alto teniendo en cuenta la temática de la aplicación. Para llegar a conseguirlo sería muy interesante realizar esta aplicación en distintos idiomas. Referencias [26 y 27].

3.4.1.2. Beneficios a través de iAd

Dentro de esta segunda posibilidad se estudia la posibilidad de utilización de la plataforma iAd, un sistema publicitario de Apple. iAd es la publicidad que se introduce dentro de las aplicaciones propias para iPhone. Está basado en contenido, por lo que los anunciantes están relacionados con la temática de la aplicación.

El anunciante tiene dos modos de pago:

- Impresión: el anunciante paga una cantidad concreta de dinero por cada mil veces que se muestre su anuncio en a aplicación.
- Click: el anunciante paga una cantidad mayor que por impresión cada vez que se hace click en el anuncio.

Los anunciantes pagan por tener sus anuncios en aplicaciones para iPhone, por lo que el usuario final de la aplicación ve reducido el coste de la aplicación. El desarrollador también sale ganando, ya que, del importe pagado por los anunciantes, se queda con un beneficio del 60% mientras que Apple se queda con el otro 40%.

Como se ha explicado anteriormente, se quiere obtener un beneficio del 20% con respecto al coste total de ambos proyectos, es decir, obtener **67.339,39€.**

Suponiendo que solamente un 1% de los usuarios que ve una impresión, pincha en el anuncio, el restante 99% entonces serán los usuarios que simplemente ven la impresión del anuncio.

Para hacer el cálculo del beneficio se necesita definir:

- X: Número de impresiones al día.

- Y: Número de días
- Z: dinero en € que paga el anunciante
- 0,99: Usuarios que sólo ven la impresión
- 0,01: Usuarios que pinchan en el anuncio
- 0,6: Beneficios del desarrollador

67.400 € = X*Y* (Z*0,99 + 0,01*1,5€) * 0,6

En la estimación de impresiones por día hay que tener en cuenta que al tratarse de una aplicación gratuita, las descargas de la aplicación se verán incrementadas. Si en el apartado anterior, se supuso una descarga diaria de la aplicación de 25 a 35 diarias, ahora se puede incrementar en un 100% ese número de descargas diarias, por lo que se rondarían entre las 50 y las 70 descargas al día. Para hacer el cálculo con una media aproximada, se suponen 60 descargas diarias.

Se debe tener en cuenta que una persona que acaba de descargarse la aplicación va a ver las impresiones más veces al día que una persona que se ha descargado la aplicación hace unos meses. Por ello, se supone que sólo el 70 % de los usuarios que tienen la aplicación realizan alguna impresión por día.

X= 60 * Y * 0,7

Si se sustituye en la primera ecuación:

67.400 € = (60*Y*0,7)*Y* (Z*0,99 + 0,01*1,5€) * 0,6

Variando el valor de la Z, se pueden obtener el número de días necesarios para conseguir una beneficio del 20%. Estos valores se pueden ver en la *Tabla 15* y de una manera más gráfica en la *Imagen 15*.

Precio por click	Precio por 1000 impresiones	Número de días para beneficio del 20%
1,5 €	2€	362 días
1,5 €	7€	251 días
1,5 €	15€	156 días
1,5 €	30€	87 días

Tabla 15 – Tiempo para obtener beneficios mediante iAds.

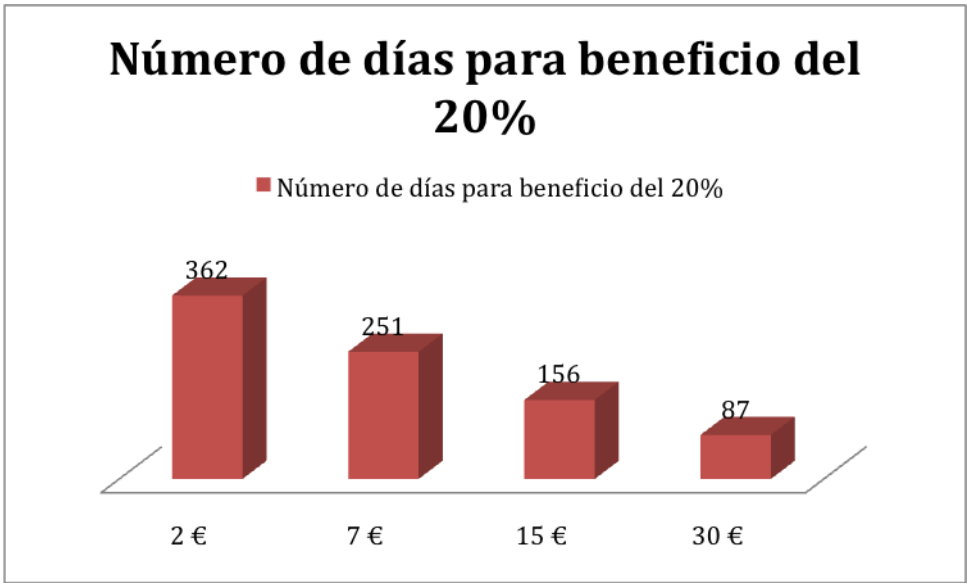


Imagen 15 – Beneficio del 20% con iAd.

Si no se utilizara el coste por click, o bien si nadie hiciera click en el anuncio, se obtendrían los resultados que se observan en la *Tabla 16* y de manera gráfica en la *Imagen 10*:

Precio por 1000 impresiones	Número de días para beneficio del 20%
2€	1338 días
7€	382 días
15€	178 días
30€	90 días

Tabla 16 – Tiempo para obtener beneficios mediante iAds sin clicks.

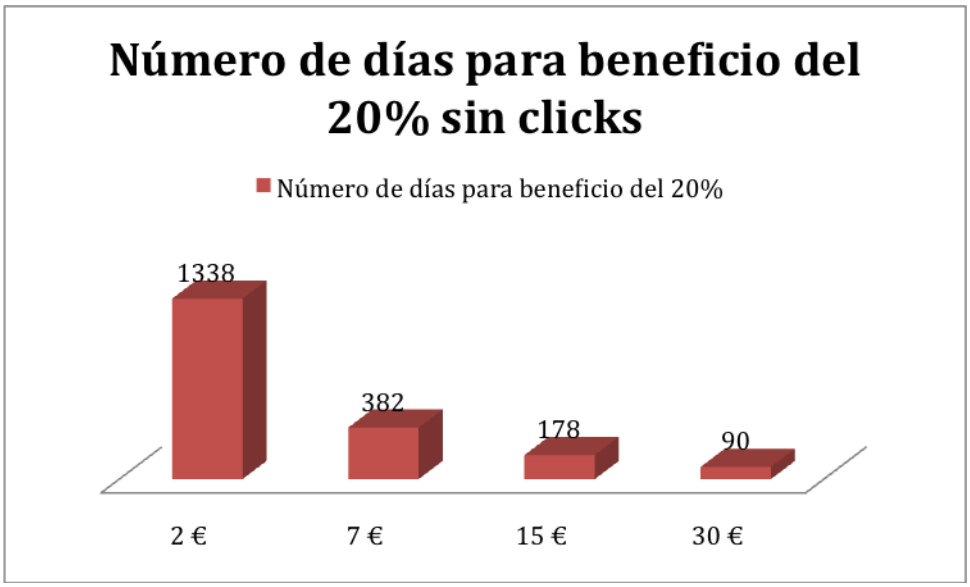


Imagen 16 – Beneficio del 20% con iAd sin clicks.

Como se puede observar comparando las *Tablas 15 y 16*, se tardan bastantes más días en obtener el beneficio esperado del 20% si no hay usuarios que accedan a la información de los anuncios.

Comparando el número de días invertidos para la obtención del beneficio, se puede ver cómo con la utilización de iAd basado en clicks más la impresión, el número de días desciende conforme se incrementa el precio por 1000 impresiones de manera lineal.

En cambio, al hacer la gráfica que aparece en la *Imagen 17*, sin utilizar clicks, solamente obteniendo ingresos por cada 1000 impresiones de pantalla, se observa cómo el descenso en días para la consecución de los beneficios sigue una distribución exponencial.

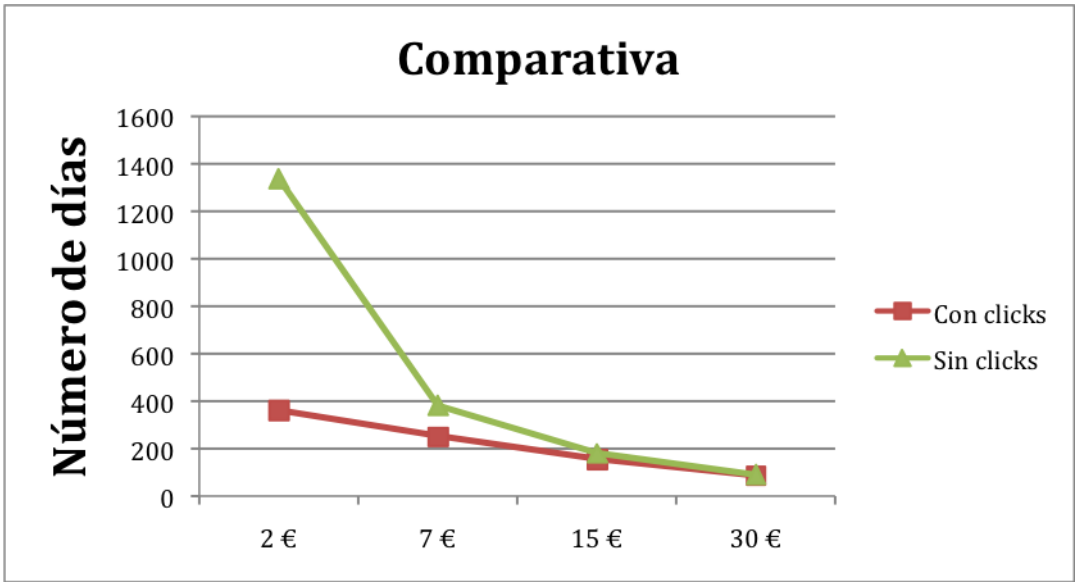


Imagen 17 – Compartiva métodos de obtener beneficios.

Siendo realistas, un anunciante pagaría 30€ por cada 1000 impresiones de pantalla si tiene el pleno convencimiento de que la aplicación en la que se pongan sus banners tiene mucha repercusión mediática. Dado el tipo de usuarios que previsiblemente utilizarán la aplicación de *Enigmatium*, es complicado que llegue a ser una aplicación que se encuentre entre las más descargadas, por lo que el anunciante presumiblemente pagaría 2€ por cada 1000 impresiones. De este modo, es mejor contar con ingresos tanto con clicks como con impresiones.

En aproximadamente 1 año (362 días), se obtendría el beneficio esperado, utilizando clicks en los anuncios, con un coste de 1.5€ por click, e impresiones de pantalla pagas a 2€ cada 1000 impresiones.

Aunque los beneficios obtenidos son mayores utilizando iAd, se optará por el método tradicional de venta no gratuita, ya que el mercado al que va dirigida la aplicación es un mercado con carácter más adquisitivo en las plataformas móviles, por lo de es bueno aprovechar este punto fuerte.

Además, iAd no está demasiado desarrollado, por lo que sería complicado en un primer momento encontrar anunciantes que estén dispuestos a invertir en *Enigmatium*. La búsqueda de anunciantes llevaría implícito una inversión de tiempo lo que se traduciría en menos tiempo de recaudación.

4. Análisis

La fase de análisis es la primera fase del ciclo de vida de este proyecto. Esta fase de análisis se ha tenido que repetir varias veces, una por cada iteración realizada, de modo que la versión final del proyecto cumpliera con los objetivos marcados.

En este apartado del documento se recogen los casos de uso, diagramas de actividad asociados a los casos de uso, requisitos de usuario y requisitos software, así como los requisitos hardware.

4.1. Captura de Requisitos

La captura de requisitos es un proceso clave en el desarrollo de un proyecto software, ya que garantiza que se conocen las especificaciones del cliente de una manera clara.

En el caso de este proyecto, los requisitos se han capturado mediante las siguientes técnicas:

- **Entrevistas con el cliente**: el analista y el cliente mantienen entrevistas tanto al inicio de una iteración como al final de la misma. De cada una de estas entrevistas se extraen los requisitos se deben formar parte del sistema.
- **Documentación:** el analista estudia el dominio del problema con el fin de recopilar la mayor documentación posible. De este modo se puede entender el dominio del problema y proponer soluciones al mismo. El analista se pone en la situación del cliente y propone soluciones que considera que son útiles para el sistema.

Siguiendo las técnicas anteriores, ha sido posible realizar un catálogo de requisitos para el sistema, así como para cada iteración.

En una entrevista inicial, se determinan los requisitos generales del sistema y las diferentes iteraciones en las que se va a dividir el mismo. Para cada una de las iteraciones, el proceso que se ha seguido en la obtención de requisitos es el siguiente:

1) Se realiza una entrevista con el cliente (el tutor en el caso de este proyecto), donde se realiza una extracción de los requisitos necesarios para la iteración.

2) Se implementan las nuevas funcionalidades.

3) Una vez concluida la implementación, se establece una nueva entrevista con el cliente donde se revisan las funcionalidades presentes en el sistema y se modifican requisitos ya definidos en caso de necesitarlo.

En caso de haber modificaciones de requisitos previos, se pasa de nuevo al punto 2.

4) Si las funcionalidades implementadas cumplen con las necesidades del cliente y con los objetivos inicialmente definidos, se valida la iteración y se pasa a la siguiente.

Para cada iteración se repiten los pasos anteriormente definidos.

4.2. Casos de Uso

En esta sección se puede observar el diagrama con los casos de uso del sistema. La inclusión de un diagrama de casos de uso facilita al analista la extracción de nuevos requisitos de usuario que en un principio no se tenían en cuenta.

Los casos de uso indican cuáles son las interacciones que el usuario realiza con la aplicación. Como se puede ver en la *Imagen 18,* el número de casos de uso del sistema es elevado, debido a la naturaleza del propio sistema, que está muy orientado a la interacción entre el usuario y la aplicación. De hecho, el único actor que posee el sistema es el usuario.

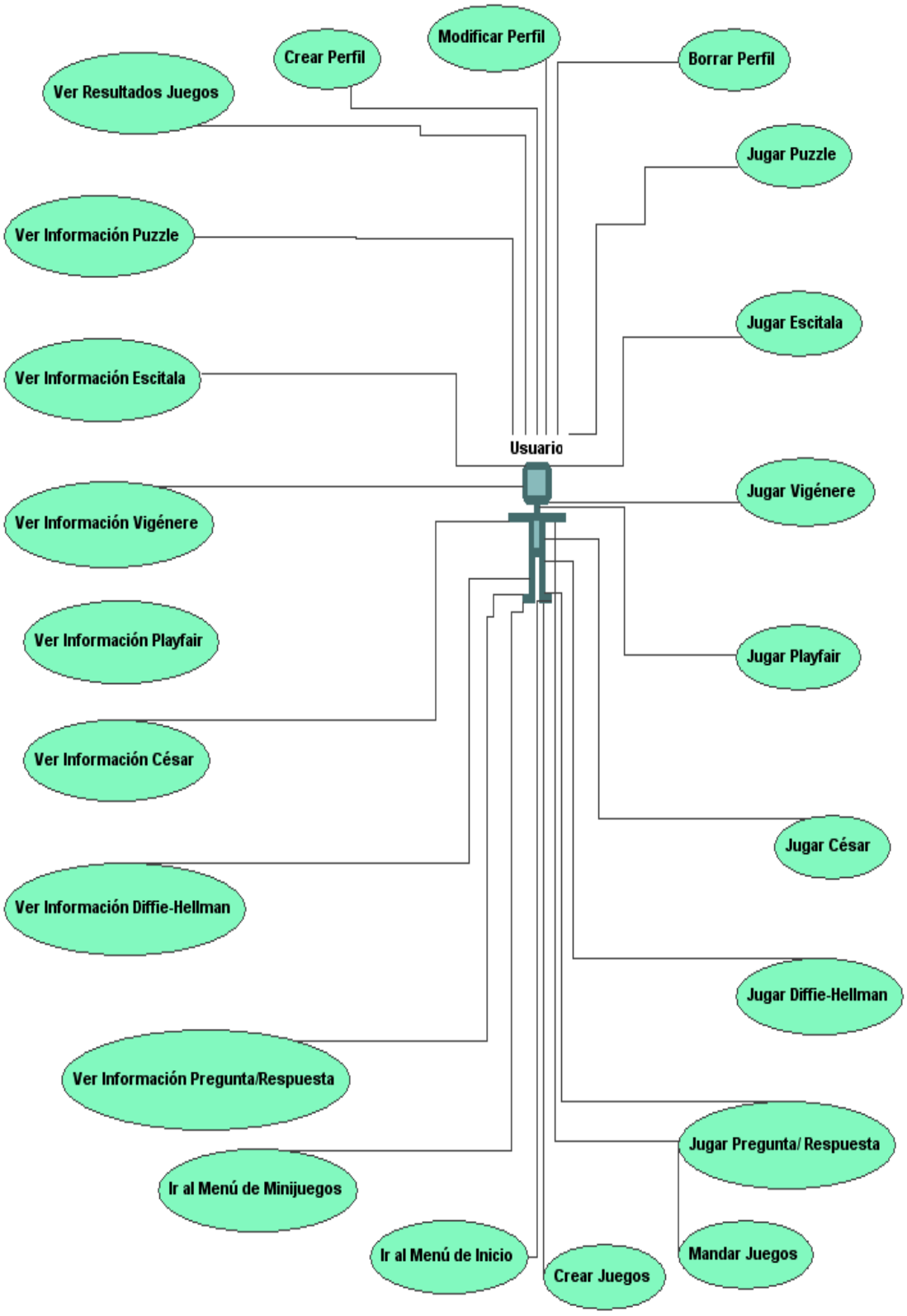


Imagen 18 – Casos de uso.

Para la especificación de los casos de uso de la *Imagen 18*, se va a utilizar la siguiente tabla para poder explicar cada caso de uso de una manera clara y concisa.

Identificador	CU.XXX
Nombre	
Descripción	
Pre-condiciones	
Flujo normal	
Post-condiciones	
Flujo alternativo	

Tabla 17 – Plantilla de casos de uso.

Cada campo de la tabla anterior, contendrá la siguiente información:

- **Identificador**: código único para cada caso de uso que lo identificará de manera inequívoca. El código debe ser especificado como se indica a continuación:
  - CU.XXX donde *CU* indica *Caso de Uso,* y *XXX* indica un número tres cifras, comenzando por el número *001* y finalizando en el *999*, de manera incremental y consecutiva.
- **Nombre**: Nombre corto y descriptivo de cada caso de uso para que el lector pueda identificar el propósito del caso de uso.
- **Descripción**: Descripción detallada del caso de uso.
- **Pre-condiciones**: Condiciones iniciales que se tienen que dar en el sistema para que pueda dar lugar a la acción relativa al caso de uso.
- **Flujo normal**: Ejecución del caso de uso, es decir, acciones que realiza el usuario.
- **Post-condiciones**: Resultados que se obtienen del sistema como consecuencia de las acciones realizadas por el usuario sobre el mismo.

- **Flujo alternativo**: Ejecución del sistema en caso de ocurrir un error en el flujo normal.

Identificador	CU.001
Nombre	Ver resultado juegos
Descripción	El sistema debe permitir al usuario acceder a los resultados obtenidos en cada juego y en cada nivel del juego.
Pre-condiciones	El usuario ha superado el nivel de algún juego.
Flujo normal	El usuario navega hasta la pantalla de los resultados.
Post- condiciones	El sistema mostrará al usuario los resultados obtenidos en cada uno de los niveles de los juegos.
Flujo alternativo	1. El usuario no ha superado ningún juego. 2. El resultado de cada nivel de los juegos será de 0 puntos.

Tabla 18 – CU.001 Ver resultados juegos.

Identificador	CU.002
Nombre	Crear perfil
Descripción	El sistema debe permitir al usuario la creación de un perfil propio.
Pre-condiciones	El usuario no ha creado un perfil con anterioridad.
Flujo normal	1. El usuario navega hasta la pantalla de perfil. 2. El usuario introduce los datos: alias, nombre, apellidos, correo electrónico e imagen.
Post-condiciones	El perfil del usuario queda almacenado en el sistema.
Flujo alternativo	El sistema muestra un error si alguno de los campos pertenecientes al perfil queda sin introducir.

Tabla 19 – CU.002 Crear perfil.

Identificador	CU.003
Nombre	Modificar perfil
Descripción	El sistema debe permitir al usuario la modificación del perfil guardado.

Pre-condiciones	El usuario tiene un perfil creado en el sistema.
Flujo normal	1. El usuario navega hasta la pantalla de perfil. 2. El usuario modifica alguno de los datos: alias, nombre, apellidos, correo electrónico o imagen.
Post-condiciones	El nuevo perfil del usuario queda almacenado en el sistema.
Flujo alternativo	El sistema muestra un error si alguno de los campos pertenecientes al perfil queda sin introducir.

Tabla 20 – CU.003 Modificar perfil.

Identificador	CU.004
Nombre	Borrar perfil
Descripción	El sistema debe permitir al usuario borrar el perfil guardado.
Pre-condiciones	El usuario tiene un perfil creado en el sistema.
Flujo normal	1. El usuario navega hasta la pantalla de perfil. 2. El usuario elimina todos los datos referentes a su perfil: alias, nombre, apellidos, correo electrónico e imagen.
Post-condiciones	El perfil del usuario queda eliminado del sistema.
Flujo alternativo	El sistema muestra un error en caso de existir algún problema al borrar el perfil.

Tabla 21 – CU.004 Borrar perfil.

Identificador	CU.005
Nombre	Ir al menú de inicio
Descripción	El sistema debe permitir al usuario acceder al menú de inicio desde cualquier parte de la aplicación.
Pre-condiciones	-
Flujo normal	El usuario selecciona la opción de ir al menú de inicio.
Post-condiciones	El sistema navega hasta el menú de inicio.
Flujo alternativo	-

Tabla 22 – CU.005 Ir al menú de inicio.

Identificador	CU.006

Nombre	Ir al menú de minijuegos
Descripción	El sistema debe permitir al usuario acceder al menú de minijuegos desde cualquier parte de la aplicación.
Pre-condiciones	-
Flujo normal	El usuario selecciona la opción de ir al menú de minijuegos.
Post-condiciones	El sistema navega hasta el menú de minijuegos.
Flujo alternativo	-

Tabla 23 – CU.006 Ir al menú de minijuegos.

Identificador	CU.007
Nombre	Ver información juego Pregunta/Respuesta
Descripción	El sistema debe permitir al usuario acceder a la información referente al juego de pregunta/respuesta.
Pre-condiciones	El usuario debe estar en la pantalla del juego correspondiente a pregunta/respuesta.
Flujo normal	1. El usuario selecciona acceder a la información del juego pregunta/respuesta. 2. El usuario navega por la información del juego de pregunta/respuesta.
Post-condiciones	1. El sistema muestra al usuario la información principal del juego de Pregunta/Respuesta. 2. El sistema muestra al usuario el resto de la información del juego de Pregunta/Respuesta.
Flujo alternativo	1. No se puede mostrar la información. 2. El sistema muestra un mensaje de error.

Tabla 24 – CU.007 Ver información juego Pregunta/Respuesta.

Identificador	CU.008
Nombre	Jugar Pregunta/Respuesta
Descripción	El sistema debe permitir al usuario jugar al juego de Pregunta/Respuesta.
Pre-condiciones	1. El usuario debe navegar hasta el juego de Pregunta/Respuesta.
Flujo normal	1. El sistema muestra el reto que corresponda en forma de pregunta.

	2. El usuario debe responder a los retos que el sistema le muestre mediante la introducción de respuestas a preguntas dadas.
Post-condiciones	1. El sistema muestra al usuario un mensaje de éxito si la pregunta es contestada correctamente. 2. El sistema muestra al usuario un mensaje de fallo en caso contrario. 3. El sistema le da la oportunidad al usuario de volver a resolver la misma pregunta.
Flujo alternativo	1. No se pueden mostrar los datos del juego. 2. El sistema muestra un mensaje de error.

Tabla 25 – CU.008 Jugar Pregunta/Respuesta.

Identificador	CU.009
Nombre	Ver información juego Diffie-Hellman
Descripción	El sistema debe permitir al usuario acceder a la información referente al juego de Diffie-Hellman.
Pre-condiciones	El usuario debe estar en la pantalla del juego correspondiente a Diffie-Hellman.
Flujo normal	1. El usuario selecciona acceder a la información del juego Diffie-Hellman. 2. El usuario navega por la información del juego de Diffie-Hellman.
Post-condiciones	1. El sistema muestra al usuario la información principal del juego de Diffie-Hellman. 2. El sistema muestra al usuario el resto de la información del juego de Diffie-Hellman.
Flujo alternativo	1. No se puede mostrar la información. 2. El sistema muestra un mensaje de error.

Tabla 26 – CU.009 Ver información juego Diffie-Hellman.

Identificador	CU.010
Nombre	Jugar Diffie-Hellman
Descripción	El sistema debe permitir al usuario jugar al juego de Diffie-Hellman.
Pre-condiciones	1. El usuario debe navegar hasta el juego de Diffie-Hellman.
Flujo normal	1. El sistema muestra los datos necesarios al usuario

	para que pueda resolver un mensaje cifrado mediante el algoritmo de Diffie-Hellman. 2. El usuario debe resolver el mensaje cifrado. 3. Para resolver el mensaje, el usuario debe introducir algunos datos que el sistema le indica y calcular otros datos también indicados.
Post-condiciones	1. El sistema muestra al usuario un mensaje de éxito si la resolución del mensaje cifrado es correcta. 2. El sistema muestra al usuario un mensaje de fallo en caso contrario. 3. El sistema le da la oportunidad al usuario de volver a resolver el mismo mensaje cifrado.
Flujo alternativo	1. No se pueden mostrar los datos del juego. 2. El sistema muestra un mensaje de error.

Tabla 27 – CU.010 Jugar Diffie-Hellman.

Identificador	CU.011
Nombre	Ver información juego César
Descripción	El sistema debe permitir al usuario acceder a la información referente al juego de César.
Pre-condiciones	El usuario debe estar en la pantalla del juego correspondiente a César.
Flujo normal	1. El usuario selecciona acceder a la información del juego César. 2. El usuario navega por la información del juego de César.
Post-condiciones	1. El sistema muestra al usuario la información principal del juego de César. 2. El sistema muestra al usuario el resto de la información del juego de César.
Flujo alternativo	1. No se puede mostrar la información. 2. El sistema muestra un mensaje de error.

Tabla 28 – CU.011 Ver información juego César.

Identificador	CU.012
Nombre	Jugar César
Descripción	El sistema debe permitir al usuario jugar al juego de César.

Pre-condiciones	1. El usuario debe navegar hasta el juego de César.
Flujo normal	1. El sistema muestra los datos necesarios al usuario para que pueda resolver un mensaje cifrado mediante el algoritmo de César. 2. El usuario debe resolver el mensaje cifrado. 3. Para resolver el mensaje, el usuario debe introducir algunos datos que el sistema le indica y calcular otros datos también indicados.
Post-condiciones	1. El sistema muestra al usuario un mensaje de éxito si la resolución del mensaje cifrado es correcta. 2. El sistema muestra al usuario un mensaje de fallo en caso contrario. 3. El sistema le da la oportunidad al usuario de volver a resolver el mismo mensaje cifrado.
Flujo alternativo	1. No se pueden mostrar los datos del juego. 2. El sistema muestra un mensaje de error.

Tabla 29 – CU.012 Jugar César.

Identificador	CU.013
Nombre	Ver información juego Playfair
Descripción	El sistema debe permitir al usuario acceder a la información referente al juego de Playfair.
Pre-condiciones	El usuario debe estar en la pantalla del juego correspondiente a Playfair.
Flujo normal	1. El usuario selecciona acceder a la información del juego Playfair. 2. El usuario navega por la información del juego de Playfair.
Post-condiciones	1. El sistema muestra al usuario la información principal del juego de Playfair. 2. El sistema muestra al usuario el resto de la información del juego de Playfair.
Flujo alternativo	1. No se puede mostrar la información. 2. El sistema muestra un mensaje de error.

Tabla 30 – CU.013 Ver información juego Playfair.

Identificador	CU.014
Nombre	Jugar Playfair

Descripción	El sistema debe permitir al usuario jugar al juego de Playfair.
Pre-condiciones	1. El usuario debe navegar hasta el juego de Playfair.
Flujo normal	1. El sistema muestra los datos necesarios al usuario para que pueda resolver un mensaje cifrado mediante el algoritmo de Playfair. 2. El usuario debe resolver el mensaje cifrado. 3. Para resolver el mensaje, el usuario debe introducir algunos datos que el sistema le indica y calcular otros datos también indicados.
Post-condiciones	1. El sistema muestra al usuario un mensaje de éxito si la resolución del mensaje cifrado es correcta. 2. El sistema muestra al usuario un mensaje de fallo en caso contrario. 3. El sistema le da la oportunidad al usuario de volver a resolver el mismo mensaje cifrado.
Flujo alternativo	1. No se pueden mostrar los datos del juego. 2. El sistema muestra un mensaje de error.

Tabla 31 – CU.014 Jugar Playfair.

Identificador	CU.015
Nombre	Ver información juego Vigenere
Descripción	El sistema debe permitir al usuario acceder a la información referente al juego de Vigenere.
Pre-condiciones	El usuario debe estar en la pantalla del juego correspondiente a Vigenere.
Flujo normal	1. El usuario selecciona acceder a la información del juego Vigenere. 2. El usuario navega por la información del juego de Vigenere.
Post-condiciones	1. El sistema muestra al usuario la información principal del juego de Vigenere. 2. El sistema muestra al usuario el resto de la información del juego de Vigenere.
Flujo alternativo	1. No se puede mostrar la información. 2. El sistema muestra un mensaje de error.

Tabla 32 – CU.015 Ver información juego Vigenere.

Identificador	CU.016

Nombre	Jugar Vigenere
Descripción	El sistema debe permitir al usuario jugar al juego de Vigenere.
Pre-condiciones	1. El usuario debe navegar hasta el juego de Vigenere.
Flujo normal	1. El sistema muestra los datos necesarios al usuario para que pueda resolver un mensaje cifrado mediante el algoritmo de Vigenere. 2. El usuario debe realizar una de las dos opciones: resolver un mensaje cifrado o bien cifrar un mensaje en claro. 3. Para ambas opciones, el usuario debe introducir algunos datos que el sistema le indica y calcular otros datos también indicados.
Post-condiciones	1. El sistema muestra al usuario un mensaje de éxito si la resolución del mensaje cifrado o el cifrado del texto claro es correcta. 2. El sistema muestra al usuario un mensaje de fallo en caso contrario. 3. El sistema le da la oportunidad al usuario de intentarlo de nuevo.
Flujo alternativo	1. No se pueden mostrar los datos del juego. 2. El sistema muestra un mensaje de error.

Tabla 33 – CU.016 Jugar Vigenere.

Identificador	CU.017
Nombre	Ver información juego Escítala
Descripción	El sistema debe permitir al usuario acceder a la información referente al juego de Escítala.
Pre-condiciones	El usuario debe estar en la pantalla del juego correspondiente a Escítala.
Flujo normal	1. El usuario selecciona acceder a la información del juego Escítala. 2. El usuario navega por la información del juego de Escítala.
Post-condiciones	1. El sistema muestra al usuario la información principal del juego de Escítala. 2. El sistema muestra al usuario el resto de la información del juego de Escítala.
Flujo alternativo	1. No se puede mostrar la información. 2. El sistema muestra un mensaje de error.

Tabla 34 – CU.017 Ver información juego Escítala.

Identificador	CU.018
Nombre	Jugar Escítala
Descripción	El sistema debe permitir al usuario jugar al juego de Escítala.
Pre-condiciones	1. El usuario debe navegar hasta el juego de Escítala.
Flujo normal	1. El sistema muestra los datos necesarios al usuario para que pueda resolver un mensaje cifrado mediante el algoritmo de la Escítala. 2. El usuario debe resolver el mensaje cifrado. 3. Para resolver el mensaje, el usuario debe introducir algunos datos que el sistema le indica.
Post-condiciones	1. El sistema muestra al usuario un mensaje de éxito si la resolución del mensaje cifrado es correcta. 2. El sistema muestra al usuario un mensaje de fallo en caso contrario. 3. El sistema le da la oportunidad al usuario de volver a resolver el mismo mensaje cifrado.
Flujo alternativo	1. No se pueden mostrar los datos del juego. 2. El sistema muestra un mensaje de error.

Tabla 35 – CU.018 Jugar Escítala.

Identificador	CU.019
Nombre	Ver información juego Puzle
Descripción	El sistema debe permitir al usuario acceder a la información referente al juego de Puzle.
Pre-condiciones	El usuario debe estar en la pantalla del juego correspondiente a Puzle.
Flujo normal	1. El usuario selecciona acceder a la información del juego Puzle. 2. El usuario navega por la información del juego de Puzle.
Post-condiciones	1. El sistema muestra al usuario la información principal del juego de Puzle. 2. El sistema muestra al usuario el resto de la información del juego de Puzle.
Flujo alternativo	1. No se puede mostrar la información. 2. El sistema muestra un mensaje de error.

Tabla 36 – CU.019 Ver información juego Puzle.

Identificador	CU.020
Nombre	Jugar Puzle
Descripción	El sistema debe permitir al usuario jugar al juego de Puzle.
Pre-condiciones	1. El usuario debe navegar hasta el juego de Puzle.
Flujo normal	1. El sistema muestra una imagen divida en pequeñas imágenes y a su vez descolocadas. 2. El usuario debe resolver la imagen y colocar las pequeñas piezas correctamente. 3. Para resolver la imagen, el usuario debe mover las piezas de lugar, hasta la obtención de la imagen original.
Post-condiciones	1. El sistema muestra al usuario un mensaje de éxito si la resolución de la imagen es correcta. 2. El sistema muestra al usuario un mensaje de fallo en caso contrario. 3. El sistema le da la oportunidad al usuario de volver a resolver la misma imagen.
Flujo alternativo	1. No se pueden mostrar los datos del juego. 2. El sistema muestra un mensaje de error.

Tabla 37 - CU.020 Jugar Puzle.

Identificador	CU.021
Nombre	Crear juegos
Descripción	El sistema debe permitir al usuario la creación de nuevos juegos.
Pre-condiciones	1. El usuario debe navegar hasta la pantalla de creación de juegos.
Flujo normal	1. El sistema se comunica con la parte de *Aventura* de *Enigmatium*. 2. La parte de *Aventura* de *Enigmatium* gestiona todo lo referente con la creación de los juegos.
Post-condiciones	1. El sistema re direcciona la petición del usuario a la zona de creación de *Aventura*.
Flujo alternativo	1. No se puede utilizar el módulo de creación. 2. El sistema muestra un mensaje de error.

Tabla 38 - CU.021 Crear juegos.

Identificador	CU.022
Nombre	Mandar juegos
Descripción	El sistema debe permitir al usuario mandar a sus amigos los juegos creados.
Pre-condiciones	1. El usuario debe navegar hasta la pantalla de mandar juegos.
Flujo normal	1. El sistema se comunica con la parte de *Aventura* de *Enigmatium*. 2. La parte de *Aventura* de *Enigmatium* gestiona todo lo referente al envío de los juegos.
Post-condiciones	1. El sistema re direcciona la petición del usuario a la zona de mandar juegos de *Aventura*.
Flujo alternativo	1. No se pueden mostrar los datos del juego. 2. El sistema muestra un mensaje de error.

Tabla 39 - CU.022 Mandar juegos.

4.3. Requisitos de Usuario

En esta sección se incluye una especificación de requisitos de usuario que el sistema debe cumplir. Estos requisitos se basan en los objetivos que se persiguen en el proyecto. Se verán ampliados dentro de la sección *Requisitos de Software*.

Para poder ofrecer una descripción clara y concisa de los requisitos de usuario, cada uno de ellos se especificará a través de la siguiente tabla:

Identificador	RU.XXX
Nombre	
Descripción	
Caso uso relacionado	

Tabla 40 – Plantilla requisitos de usuario.

Cada campo de la tabla anterior, contendrá la siguiente información:

- **Identificador**: código único para cada requisito de usuario que lo identificará de manera inequívoca. El código debe ser especificado como se indica a continuación:
    - RU.XXX donde *RU* indica *Requisito de Usuario,* y *XXX* indica un número tres cifras, comenzando por el número *001* y finalizando en el *999*, de manera incremental y consecutiva.
- **Nombre**: Nombre corto y descriptivo de cada requisito de usuario para que el lector pueda identificar fácilmente el propósito del requisito de usuario.
- **Descripción**: Descripción detallada del requisito de usuario.
- **Caso uso relacionado**: Caso de uso relacionado con el requisito de usuario o "-" en caso de no tener un caso de uso relacionado.

Identificador	RU.001
Nombre	Contenido de la aplicación
Descripción	El sistema debe contener ejercicios basados en algoritmos criptográficos y de seguridad tecnológica.
Caso uso relacionado	CU.008, CU.010, CU.012, CU.014, CU.016, CU.018, CU.020.

Tabla 41 – RU.001 Contenido de la aplicación.

Identificador	RU.002
Nombre	Recurso docente
Descripción	La aplicación debe utilizar ejercicios en forma de minijuegos.
Caso uso relacionado	CU.007,CU.008,CU.009,CU.010,CU.011,CU.012, CU.013,CU.014,CU.015,CU.016,CU.017,CU.018, CU.019,CU.020.

Tabla 42 – RU.002 Recurso docente.

Identificador	RU.003
Nombre	Diversidad de juegos
Descripción	El sistema debe contener un conjunto de juegos diferentes entre ellos basado en un algoritmo criptográfico diferente.

Caso uso relacionado	CU.008,CU.010,CU.012,CU.014,CU.016,CU.018, CU.020.

Tabla 43 – RU.003 Diversidad de juegos.

Identificador	RU-004
Nombre	Niveles de dificultad de los juegos
Descripción	Cada juego dispondrá de diferentes niveles de dificultad.
Caso uso relacionado	CU.008,CU.010,CU.012,CU.014,CU.016,CU.018, CU.020.

Tabla 44 – RU.004 Niveles de dificultad de los juegos.

Identificador	RU.005
Nombre	Información sobre cada juego
Descripción	El sistema debe proporcionar información acerca de: 1. Algoritmo de juego. 2. Funcionalidad del sistema en sí mismo. 3. Método de puntuación para cada juego.
Caso uso relacionado	CU.007,CU.009,CU.011,CU.013,CU.015,CU.017, CU.019.

Tabla 45 – RU.005 Información sobre cada juego.

Identificador	RU-006
Nombre	Juegos a implementar
Descripción	Los juegos que se van a implementar en el sistema son: 1. Puzle 2. Escítala 3. César 4. Vigenere 5. Playfair 6. Diffie-Hellman 7.Pregunta / Respuesta
Caso uso relacionado	CU.008,CU.010,CU.012,CU.014,CU.016,CU.018, CU.020.

Tabla 46 – RU.006 Juegos a implementar.

Identificador	RU-007

Nombre	Consulta de los resultados obtenidos
Descripción	El sistema debe permitir al usuario el acceso a los resultados obtenidos desde el menú correspondiente a resultados.
Caso uso relacionado	CU.001

Tabla 47 – RU.007 Consulta de los resultados obtenidos.

Identificador	RU-008
Nombre	Personalización de la aplicación
Descripción	El sistema debe ser personalizable, de manera que permita al usuario introducir datos de carácter personal. Así información del sistema quedará relacionada con el usuario.
Caso uso relacionado	CU.002, CU.003, CU.004

Tabla 48 – RU.008 Personalización de la aplicación.

4.4. Explicación de los juegos

En esta sección se explica para cada juego presente en la aplicación, el algoritmo que siguen así como la implementación que se hace de cada uno de estos algoritmos.

4.4.1. J1 - César

Descripción

El cifrado César, también llamado de desplazamiento se basa en un cifrado por sustitución. La sustitución consiste en el reemplazamiento de una letra del texto original por otra letra que se encuentra un número fijo de posiciones más adelante en el alfabeto:

- Si el desplazamiento o sustitución es por una letra que se encuentra tres posiciones exactas más adelante en el alfabeto, se llama cifrado César.
- Si el desplazamiento o sustitución es por una letra que se encuentra más de tres posiciones adelante en el alfabeto, se llama cifrado por desplazamiento.

En este tipo de cifrado se usa el mismo procedimiento tanto para cifrar como para descifrar.

Datos de entrada

1. Para realizar un cifrado:

- Mensaje en claro

- Desplazamiento a aplicar

2. Para realizar un descifrado:

- Mensaje cifrado

- Desplazamiento a aplicar

Algoritmo de resolución

1. Para realizar un cifrado: dado el texto en claro, se utiliza el desplazamiento dado y se aplica la siguiente operación matemática a cada letra del texto: C = (x + n) mód 27, siendo

C: letra cifrada

x: letra original

n: número de desplazamiento

27: número de letras del alfabeto utilizado

2. Para realizar un descifrado: dado el texto cifrado, se utiliza el desplazamiento dado y se aplica la siguiente operación matemática a cada letra del texto: X = (C - n) mód 27, siendo

X: letra descifrada

C: letra cifrada

n: número de desplazamiento

27: número de letras del alfabeto utilizado

Datos de salida

1. Para realizar un cifrado:

- Mensaje cifrado.

2. Para realizar un descifrado:

- Mensaje en claro.

Dinámica del juego

El cifrado césar se implementa en el sistema de la siguiente manera: el sistema mostrará al usuario cuál es el desplazamiento a utilizar en el juego así como el mensaje cifrado. Así mismo el sistema proporcionará al usuario un

mecanismo de ayuda que le permita establecer la correspondencia entre la letra cifrada y la letra en claro, dependiendo del desplazamiento aplicado. Este mecanismo de ayuda se implementa a través de una ruleta donde aparezca una letra, correspondiente con la letra cifrada, y a su lado se encuentra otra letra, que se corresponde con la letra en claro.

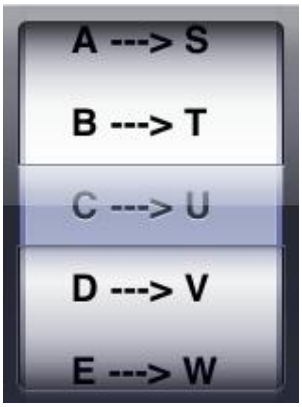


Imagen 19 – Ayuda César.

El usuario debe buscar la correspondencia entre cada una de las letras cifradas con su respectiva letra en claro. Cuando el usuario finalice la correspondencia entre las letras, podrá comprobar si la solución introducida es correcta.

4.4.2. J2 - Vigenere

DESCRIPCIÓN

El cifrado de Vigenere es un cifrado polialfabético y de sustitución basado en diferentes series de caracteres o letras.

DATOS DE ENTRADA

1. Para realizar un cifrado:

- Mensaje en claro
- Clave a aplicar

2. Para realizar un descifrado:

- Mensaje cifrado
- Clave a aplicar

Algoritmo de resolución

1. Para realizar un cifrado: dado el texto en claro, se utiliza la clave para obtener el texto cifrado de la siguiente forma:

M: mensaje en claro

C: mensaje cifrado

x_i: valor de la letra i del mensaje en claro

y_i: valor de la letra i de la clave a utilizar

c_i: valor de la letra i del mensaje cifrado

27: número de letras del alfabeto utilizado

M = MENSAJE CLARO

$$\begin{array}{r} x_1\ x_2\ x_3\ x_4\ x_5\ \ldots\ldots\ x_n \\ +\ y_1\ y_2\ y_3\ y_4\ y_5\ \ldots\ldots\ y_n \quad \text{mód } 27 \\ \hline c_1\ c_2\ c_3\ c_4\ c_5\ \ldots\ldots\ c_n \end{array}$$

C = MENSAJE CIFRADO

2. Para realizar un descifrado: dado el texto cifrado, se utiliza la clave para obtener el texto claro de la siguiente forma:

M: mensaje en claro

C: mensaje cifrado

x_i: valor de la letra i del mensaje en claro

y_i: valor de la letra i de la clave a utilizar

c_i: valor de la letra i del mensaje cifrado

27: número de letras del alfabeto utilizado

C = MENSAJE CIFRADO

$$\begin{array}{r} c_1\ c_2\ c_3\ c_4\ c_5\ \ldots\ldots\ c_n \\ +\ y_1\ y_2\ y_3\ y_4\ y_5\ \ldots\ldots\ y_n \quad \text{mód } 27 \\ \hline x_1\ x_2\ x_3\ x_4\ x_5\ \ldots\ldots\ x_n \end{array}$$

M = MENSAJE CLARO

Datos de salida

1. Para realizar un cifrado:

- Mensaje cifrado.

2. Para realizar un descifrado:

- Mensaje en claro.

Dinámica del juego

La resolución del juego sigue el algoritmo de cifrado de Vigenere con pequeñas modificaciones. La clave a utilizar puede ser de dos tipos:

1. Colocación clave normal: si el texto es más largo que el tamaño de la clave, se repite la clave tantas veces como sea necesario.

Imagen 20 – Vigenere con clave normal.

2. Autoclave: si el texto es más largo que el tamaño de la clave, se coloca la clave una sola vez y a continuación se van colocando las letras que se obtienen de la codificación inicial.

Imagen 21 – Vigenere con autoclave.

El sistema mostrará al usuario cuál es la clave a utilizar en el juego y el mensaje cifrado. Así mismo se le proporcionará al usuario un mecanismo de ayuda para que éste pueda realizar las operaciones necesarias para obtener el mensaje final. Esta ayuda será en forma de ruleta, y para realizar los cálculos sólo tendrá que colocar las letras correctamente.

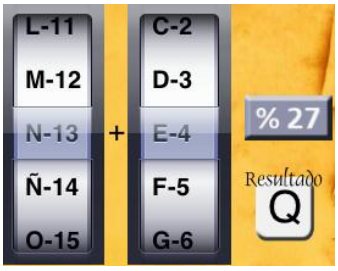


Imagen 22 – Ayuda Vigenere.

Cuando el usuario finalice la correspondencia entre las letras, podrá comprobar si la solución introducida es correcta.

4.4.3. J3 - Playfair

Descripción

En el cifrado de Playfair, un par de letras sin codificar corresponden a otro par de letras diferente codificado.

Datos de entrada

1. Para realizar un cifrado:

- Mensaje en claro

- Clave a aplicar

2. Para realizar un descifrado:

- Mensaje cifrado

- Clave a aplicar

Algoritmo de resolución

1. Se coloca la clave en una matriz 5x5, sin repetir ninguna letra (comenzando por la posición (0,0) y terminando en la (4,4)).

2. Se coloca el resto del abecedario, sin repetir letras ya colocadas, teniendo en cuenta lo siguiente:

- La J se coloca en la misma casilla que la I.

- La Ñ se coloca en la misma casilla que la N.

3. Para cifrar: dado el texto en claro, se cogen pares de letras (m1, m2) y se codifican de la siguiente manera:

a. Si m1 y m2 están en la misma fila: coger c1 c2 a su derecha (circularmente).

b. Si m1 y m2 están en la misma columna: coger c1 c2 de debajo (circularmente).

c. Si m1 y m2 están en distintas filas y columnas, coger c1 c2 de la diagonal opuesta.

d. Si m1 = m2, meter un carácter de relleno entre ambas para evitar su repetición y codificar siguiendo a,b,c.

e. Si el número de letras es impar, introducir un carácter de relleno al final.

4. Para descifrar: dado el texto en cifrado, se cogen pares de letras (m1, m2) y se resuelven de la siguiente manera:

a. Si m1 y m2 están en la misma fila: coger c1 c2 a su izquierda (circularmente).

b. Si m1 y m2 están en la misma columna: coger c1 c2 de encima (circularmente).

c. Si m1 y m2 están en distintas filas y columnas, coger c1 c2 de la diagonal opuesta.

d. Si m1 = m2, meter un carácter de relleno entre ambas para evitar su repetición y codificar siguiendo a,b,c.

e. Si se obtiene un carácter fuera de contexto, ignorarlo, es un carácter de relleno al final.

Datos de salida

1. Para realizar un cifrado:

- Mensaje cifrado.

2. Para realizar un descifrado:

- Mensaje en claro.

Dinámica del juego

La resolución del juego sigue el algoritmo de cifrado de Playfair con pequeñas modificaciones.

La clave a utilizar se puede ser de dos maneras:

1. Colocación clave normal: la clave se coloca comenzando en el extremo superior izquierdo de la matriz (posición (0,0)) y finalizando en el extremo inferior derecho de la matriz (posición (4,4)).

C	L	A	V	E
B	D	F	G	H
I	K	M	N	O
P	Q	R	S	T
U	W	X	Y	Z

Imagen 23 – Colocación clave normal Playfair.

2. Colocación clave en espiral: la clave se coloca comenzando en la casilla del centro de la matriz (posición (2,2)).

M	N	O	P	Q
K	A	V	E	R
I	L	C	B	S
H	G	F	D	T
Z	Y	X	W	U

Imagen 24 – Colocación clave espiral Playfair.

3. La colocación de las letras en la matriz seguirá el siguiente orden: (2,2), (2,1), (1,1), (1,2), (1,3), (2,3), (3,3), (3,2), (3,1), (3,0), (2,0), (1,0) ,(0,0), (0,1), (0,2), (0,3), (0,4), (1,4), (2,4), (3,4), (4,4), (4,3), (4,2), (4,1), (4,0).

El sistema mostrará al usuario la clave a utilizar en el juego y el mensaje cifrado. Se proporcionará una matriz para que el usuario coloque las letras, así como un mecanismo de ayuda que coloque la clave en la matriz. El usuario deber terminar de colocar el resto de letras del abecedario en la matriz, recordando las reglas que debe seguir para ello.

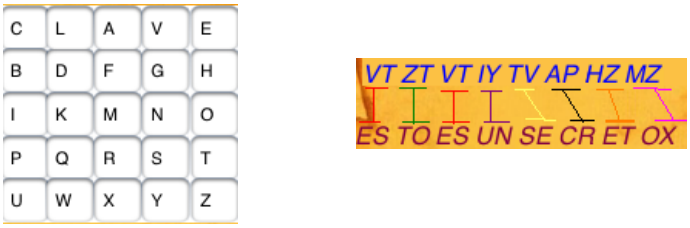


Imagen 25 – Resolución Playfair.

El usuario debe realizar los cálculos necesarios, con la ayuda que le proporciona el sistema, para obtener la correspondencia entre el mensaje cifrado y el mensaje en claro. Cuando el usuario finalice la correspondencia entre las letras, podrá comprobar si la solución introducida es correcta.

4.4.4. J4 - Escítala

Descripción

Una Escítala es un sistema de codificación utilizada por los espartanos para el envío de mensajes secretos. Se basa en la trasposición más elemental seriada y continua (no tiene saltos).

Datos de entrada

1. Para realizar un cifrado:
   - Mensaje en claro
   - Tamaño de la Escítala
2. Para realizar un descifrado:
   - Mensaje cifrado
   - Tamaño de la Escítala

Algoritmo de resolución

1. Para realizar un cifrado: dado el texto en claro, disponer en una tabla cada uno de los elementos en filas y luego tomarlos en columnas.

2. Para realizar un descifrado: dado el texto cifrado, disponer en una tabla cada uno de los elementos en columnas y luego tomarlos en filas.

El ancho de la fila representa el número de caras que presenta la Escítala y el número de filas la cantidad resultante de dividir el largo total del mensaje entre el ancho de la fila.

Datos de salida

1. Para realizar un cifrado:

- Mensaje cifrado.

2. Para realizar un descifrado:

- Mensaje en claro.

Dinámica del juego

La resolución del juego sigue el algoritmo de cifrado de Escítala con una pequeña modificación, no se le indicará al usuario el tamaño de la Escítala, por lo que tendrá que probar entre los diferentes tamaños hasta ver cuál es el que se ajusta mejor al tamaño del mensaje cifrado.

El sistema muestra al usuario el mensaje cifrado y le proporciona al usuario una matriz para que coloque las letras del mensaje cifrado. El usuario deber colocar el mensaje cifrado en la matriz.

Imagen 26 – Colocación mensaje Escítala.

El usuario debe cambiar el tamaño de la Escítala (representado por la matriz) si así lo considera oportuno y establecer la correspondencia entre el mensaje cifrado y el mensaje en claro. Cuando el usuario finalice la correspondencia entre las letras, podrá comprobar si la solución introducida es correcta.

4.4.5. J5 - Diffie-Hellman

Descripción

El protocolo Diffie-Hellman permite el intercambio de claves entre dos partes que no han tenido contacto previo, utilizando un canal inseguro y de manera anónima (no autenticada).

Datos de entrada

1. Número primo *p*

2. Número *g*, generador del número primo *p*.

Para cada una de las partes:

a: clave privada de una de las partes.

b: clave privada de una de las partes.

Algoritmo de resolución

Se tienen dos personas, "X" e "Y". Dados los números *p* (número primo muy grande) y *g* (generador del número primo).

1. "X" elige una clave privada *a* y calcula su clave pública: $A = g^a$ mód p

2. "X" envía a "Y" *A, p y g*. "Y" elige una clave privada b y calcula su clave pública: $B = g^b$ mód p

3. "Y" envía a "X" su clave pública B.

4. "Y" calcula la clave compartida: $K = A^b$ mód p

5. "X" calcula la clave compartida: $K = B^a$ mód p

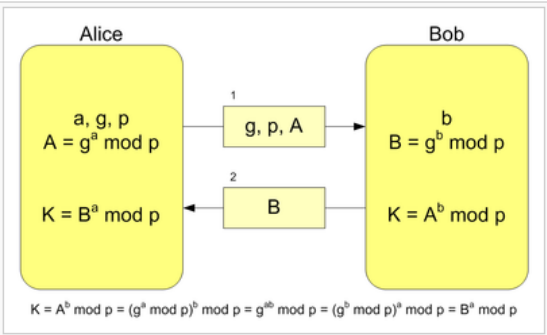


Imagen 27 – Cálculo parámetros Diffie-Hellman [34].

Datos de salida

Clave *K* compartida por ambas partes.

Dinámica del juego

La resolución del juego sigue el protocolo de intercambio de claves de Diffie-Hellman con un detalle añadido: además de realizarse un intercambio de claves, el usuario tendrá que descifrar el mensaje recibido.

El sistema muestra al usuario los números *p* y *g* y el usuario introduce una clave privada. El usuario calcula su clave pública en función de la clave privada elegida.

El sistema a su vez muestra al usuario la clave pública recibida de otro usuario por lo que el usuario debe calcular la clave compartida. Una vez que tiene la clave compartida, tendrá que resolver el mensaje cifrado recibido utilizando la clave compartida que acaba de calcular.

El mensaje recibido se cifra utilizando la clave K a modo de desplazamiento (para cifrar se utiliza el cifrado por desplazamiento, el desplazamiento aplicado es K), por lo que para descifrarlo tendrá que calcular:

X = (C - K) mód 27

4.4.6. J6 - Puzle

Descripción

El algoritmo N-Puzle consiste en una imagen dividida en pequeños cuadrados, con uno de ellos libre y descolocado.

Datos de entrada

Fragmentos de la imagen dividida colocados de manera aleatoria en una matriz NxN.

Algoritmo de resolución

Con la imagen dividida en pequeños fragmentos y colocados de manera aleatoria, se moverán los fragmentos que estén situados próximos a la única casilla en blanco. Sólo se puede mover un fragmento a la vez. Dependiendo de la colocación inicial de los fragmentos, es posible que el juego no tenga solución.

Datos de salida

Fragmentos de la imagen colocados de manera ordenada dentro de la matriz NxN con una imagen bien definida.

Dinámica del juego

La resolución del juego sigue el algoritmo N-Puzle particularizando en el 9-Puzle.

El sistema muestra al usuario los fragmentos de una imagen colocados de manera aleatoria, de manera que debe mover de una en una las piezas del Puzle hasta que consigue obtener la imagen inicial. Una vez la imagen está colocada correctamente, el usuario introduce el nombre que le sugiere la imagen.

4.4.7. J7 - Pregunta / Respuesta

Descripción

El juego de pregunta/respuesta consiste en una serie de preguntas propuestas al usuario para que las resuelva. Este juego está basado en los juegos de mesa más conocidos: Trivial (donde se hace una pregunta y se proponen varias respuesta), Tabú (donde se dan pista mediante varias palabras y hay que averiguar el resultado), Pictionary (donde a través de un dibujo hay que averiguar de qué se trata). Y finalmente se ha añadido un juego llamado Contador que se basa en introducir 5 ejemplos de un tema dado en un tiempo límite. Pasado este tiempo no se puede seguir escribiendo.

Datos de entrada

Preguntas teóricas referentes a temas relacionados con la criptografía y la seguridad en las tecnologías de la información.

Algoritmo de resolución

Este juego no sigue ningún algoritmo criptográfico de resolución.

Datos de salida

Respuesta o respuestas a la pregunta propuesta.

Dinámica del juego

El sistema muestra al usuario unas preguntas cuya estructura y dinámica se basan en los juegos de mesa: Trivial, Tabú, Pictionary y uno llamado Contador.

En el juego de Trivial, el sistema muestra una pregunta, ofrece una serie de respuestas al usuario entre las cuales se encuentra la solución. El usuario debe seleccionar una de las opciones.

En el juego de Tabú, el sistema muestra al usuario cuatro palabras de manera que el usuario debe introducir qué le sugieren.

En el juego de Pictionary, el sistema muestra al usuario cuatro palabras y una imagen y el usuario debe introducir qué le sugieren las palabras y la imagen.

En el juego de Contador, el sistema muestra una pregunta o tema y el usuario debe introducir cinco respuestas a la pregunta propuesta. El usuario cuenta con un tiempo limitado e indicado para introducir las respuestas.

4.5. Requisitos de Software

En esta sección se incluye una especificación de requisitos de software que el sistema debe cumplir. Dentro de esta especificación de requisitos software no se incluye una especificación típica, ya que para poder explicar cada uno de los juegos que forman parte del sistema, se debe hacer en un apartado diferente.

Para poder ofrecer una descripción clara y concisa de los requisitos de software, cada uno de ellos se especificará a través de la siguiente tabla:

Identificador	RS.F/NF.XXX.YY	**Tipo Requisito**	
Nombre			
Descripción			
Dependencias			
Procedencia			

Tabla 49 – Plantilla requisitos de software.

Cada campo de la tabla anterior, contendrá la siguiente información:

- **Identificador**: código único para cada requisito de software que lo identificará de manera inequívoca. El código debe ser especificado como se indica a continuación:
    - RS.F.XXX.YY donde *RS* indica *Requisito Software, F* indica requisito *Funcional*, *XXX* indica requisito de usuario del que depende e *YY* indica es un número de dos dígitos comenzando por el número *01* y finalizando en el *99*, de manera incremental y consecutiva.
    - RS.NF.XXX.YY donde *RS* indica *Requisito Software, NF* indica requisito *No Funcional*, *XXX* indica requisito de usuario del que depende e *YY* indica es un número de dos dígitos comenzando por el número *01* y finalizando en el *99*, de manera incremental y consecutiva.

- **Tipo de requisito:** Tipo de requisito de software. Los posibles valores son: “Funcional”, “Arquitectura”, “Seguridad”, “Usabilidad” e “Interfaz”. Excluyendo el tipo “Funcional”, el resto se corresponde con tipos de requisitos no funcionales.
- **Nombre**: Nombre corto y descriptivo de cada requisito de software para que el lector pueda identificar el propósito del requisito de software.
- **Descripción**: Descripción detallada del requisito de software.
- **Dependencias**: Requisito de software que se ven afectados o afectan al requisito que se describe. Si no es aplicable, se indica mediante “-”.
- **Procedencia**: Requisito de usuario del que precede el requisito de software que se describe. Si no es aplicable, se indica mediante “-”.

4.5.1. Requisitos funcionales

En esta sección se describen los requisitos de software que describen las funcionalidades que el sistema debe ofrecer:

Identificador	RS.F.001.01	**Tipo Requisito**	Funcional
Nombre	César		
Descripción	El sistema debe incluir un juego que implemente el cifrado de César, tal como se explica en el apartado J1 - César.		
Dependencias	RS.F.001.01, RS.F.003.02		
Procedencia	RU.001		

Tabla 50 – RS.F.001.01 César.

Identificador	RS.F.001.02	**Tipo Requisito**	Funcional
Nombre	Vigenere		
Descripción	El sistema debe incluir un juego que implemente el cifrado de Vigenere, tal como se explica en el apartado J2 - Vigenere.		
Dependencias	RS.F.001.01, RS.F.003.03		
Procedencia	RU.001		

Tabla 51 – RS.F.001.02 Vigenere.

Identificador	RS.F.001.03	Tipo Requisito	Funcional
Nombre	Playfair		
Descripción	El sistema debe incluir un juego que implemente el cifrado de Playfair, tal como se explica en el apartado J3 - Playfair.		
Dependencias	RS.F.001.01, RS.F.003.04		
Procedencia	RU.001		

Tabla 52 – RS.F.001.03 Playfair.

Identificador	RS.F.001.04	Tipo Requisito	Funcional
Nombre	Escítala		
Descripción	El sistema debe incluir un juego que implemente el cifrado de Escítala, tal como se explica en el apartado J4 - Escíta-la.		
Dependencias	RS.F.001.01, RS.F.003.05		
Procedencia	RU.001		

Tabla 53 – RS.F.001.04 Escítala.

Identificador	RS.F.001.05	Tipo Requisito	Funcional
Nombre	Diffie-Hellman		
Descripción	El sistema debe incluir un juego que implemente el cifrado de Diffie-Hellman, tal como se explica en el apartado J5 - Diffie-Hellman.		
Dependencias	RS.F.001.01, RS.F.003.06		
Procedencia	RU.001		

Tabla 54 – RS.F.001.05 Diffie-Hellman.

Identificador	RS.F.001.06	Tipo Requisito	Funcional
Nombre	Puzle		
Descripción	El sistema debe incluir un juego que implemente el cifrado de Puzle, tal como se explica en el apartado J6 - Puzle.		
Dependencias	RS.F.001.01, RS.F.003.07		
Procedencia	RU.001		

Tabla 55 – RS.F.001.06 Puzle.

Identificador	RS.F.001.07	**Tipo Requisito**	Funcional
Nombre	Pregunta/Respuesta		
Descripción	El sistema debe incluir un juego que implemente el cifrado de Pregunta/Respuesta, tal como se explica en el apartado *J7 - Pregunta / Respuesta*.		
Dependencias	RS.F.001.01, RS.F.003.07		
Procedencia	RU.001		

Tabla 56 – RS.F.001.07 Pregunta/Respuesta.

Identificador	RS.F.003.01	**Tipo Requisito**	Funcional
Nombre	Bloqueo de los juegos		
Descripción	Según el nivel en el que se encuentre el usuario, se encontrarán bloqueados un número de niveles. Si el usuario no ha superado un nivel, el siguiente nivel no estará desbloqueado. De manera que el usuario sólo podrá jugar al nivel que tiene sin superar y a los niveles inferiores al mismo.		
Dependencias	RS.F.003.01, RS.F.003.2		
Procedencia	RU.003		

Tabla 57 – RS.F.003.01 Niveles de los juegos.

Identificador	RS.F.003.2	**Tipo Requisito**	Funcional
Nombre	Puntuaciones de los juegos		
Descripción	La puntuación se guardará en el sistema después de la resolución de cada minijuego.		
Dependencias	RS.F.003.01		
Procedencia	RU.003		

Tabla 58 – RS.F.003.2 Puntuaciones de los juegos.

Identificador	RS.F.004.01	**Tipo Requisito**	Funcional
Nombre	Información sobre los juegos		
Descripción	El sistema debe proporcionar información acerca de: 1. Algoritmo de juego. 2. Funcionalidad del sistema en sí mismo. 3. Método de puntuación para cada juego. El usuario podrá consultar la información relativa a cada		

	juego tantas veces como desee.
Dependencias	RS.F.004.01
Procedencia	RU.004

Tabla 59 – RS.F.004.01 Información sobre los juegos.

Identificador	RS.F.006.01	**Tipo Requisito**	Funcional
Nombre	Acceder al menú de inicio		
Descripción	El sistema debe permitir al usuario acceder al menú inicio desde cualquier parte de la aplicación. 1. El usuario puede acceder al menú de inicio independientemente del lugar del sistema donde se encuentre. 2. Desde el menú de inicio se permite el acceso tanto a las opciones del sistema como al proyecto complementario al que se está describiendo, llamado *Aventura.*		
Dependencias	-		
Procedencia	RU.006		

Tabla 60 – RS.F.006.01 Acceder al menú de inicio.

Identificador	RS.F.006.02	**Tipo Requisito**	Funcional
Nombre	Acceder al menú de minijuegos		
Descripción	El sistema debe permitir al usuario acceder al menú de minijuegos desde cualquier parte de la aplicación: 1. El usuario puede acceder al menú de minijuegos independientemente del lugar del sistema donde se encuentre. 2. Desde el menú de minijuegos se permite al usuario el acceso tanto al resto de juegos.		
Dependencias	-		
Procedencia	RU.006		

Tabla 61 – RS.F.006.02 Acceder al menú de minijuegos.

Identificador	RS.F.007.01	**Tipo Requisito**	Funcional
Nombre	Acceder al menú de resultados		
Descripción	El sistema debe permitir al usuario acceder al menú de resultados desde cualquier parte de la aplicación.		

	1. El usuario puede acceder al menú de resultados independientemente del lugar del sistema donde se encuentre. 2. En el menú de resultados, el usuario podrá consultar: - Juegos del sistema. - Niveles de cada juego. - Puntos de cada nivel.
Dependencias	RS.F.003.02, RS.F.003.03, RS.F.003.04, RS.F.003.05, RS.F.003.06, RS.F.003.07, RS.F.003.08, RS.F.003.09, RS.F.003.10
Procedencia	RU.007

Tabla 62 – RS.F.007.01 Acceder al menú de resultados.

Identificador	RS.F.008.01	**Tipo Requisito**	Funcional
Nombre	Acceder al menú de perfil		
Descripción	El sistema debe permitir al usuario acceder al menú de perfil desde cualquier parte de la aplicación. 1. El usuario puede acceder al menú de perfil desde el menú de inicio. 2. Desde el menú de perfil, el usuario podrá realizar las siguientes acciones: - Crear nuevo perfil: introduciendo el alias, nombre, apellidos, correo electrónico e imagen. - Modificar perfil existente: modificando el alias, nombre, apellidos, correo electrónico o imagen, o bien todas ellas. - Eliminar perfil existente.		
Dependencias	RS.F.006.01		
Procedencia	RU.008		

Tabla 63 – RS.F.008.01 Acceder al menú de perfil.

4.5.2. Requisitos no funcionales

En esta sección se describen los requisitos de software que describen las restricciones que el sistema debe cumplir:

Identificador	RS.NF.003.01	**Tipo Requisito**	
Nombre	Niveles de dificultad		
Descripción	Cada juego dispondrá de diferentes niveles de dificultad.		

Dependencias	RS.F.003.01, RS.F.003.2
Procedencia	RU.003

Tabla 64 – RS.NF.003.01 Niveles de dificultad.

Identificador	RS.NF.003.2	**Tipo Requisito**	
Nombre	Puntuaciones de los juegos		
Descripción	Cada nivel de juego tiene una puntuación independiente del resto de niveles del juego. 1. Al comenzar a jugar en cada uno de los niveles, la puntuación se le asigna al usuario es de 100 puntos. 2. Si el usuario no resuelve el juego de manera correcta, se le restan 20 puntos. 3. El usuario puede seguir intentando resolver el juego. 1. 4. Cuando el usuario llega al intento número 6 o más, la puntuación que obtendrá a partir de este intento será sólo de 10 puntos.		
Dependencias	RS.F.003.01, RS.F.003.09		
Procedencia	RU.003		

Tabla 65 – RS.NF.003.2 Puntuaciones de los juegos.

Identificador	RS.NF.005.01	**Tipo Requisito**	Arquitectura
Nombre	Implementación modular del sistema		
Descripción	El sistema debe ser diseñado de tal manera que no suponga un esfuerzo añadido el incluir nuevas funcionalidades sin perjudicar a las ya disponibles en el sistema.		
Dependencias	-		
Procedencia	RU.005		

Tabla 66 – RS.NF.005.01 Implementación modular.

4.6. Requisitos de Hardware

En esta sección se incluye una especificación de requisitos de hardware necesarios para que el sistema pueda funcionar correctamente.

Para poder ofrecer una descripción clara y concisa de los requisitos de hardware, cada uno de ellos se especificará a través de la siguiente tabla:

Identificador	RH.XXX
Nombre	
Descripción	

Tabla 67 – Plantilla requisitos de hardware.

Cada campo de la tabla anterior, contendrá la siguiente información:

- **Identificador**: código único para cada requisito de hardware que lo identificará de manera inequívoca. El código debe ser especificado como se indica a continuación:
  - RH.XXX. donde *RH* indica *Requisito Hardware* y *YY* es un número de dos dígitos comenzando por el número *01* y finalizando en el *99*, de manera incremental y consecutiva.
- **Nombre**: Nombre corto y descriptivo de cada requisito de hardware para que el lector pueda identificar el propósito del requisito de hardware.
- **Descripción**: Descripción detallada del requisito de hardware.

Identificador	RH.01
Nombre	Tipo de dispositivo
Descripción	La aplicación está orientada a instalarse en dispositivos móviles de Apple, en concreto: 1. iPhone 4 2. iPad 1 y iPad 2 3. iPod Touch 3G No se garantiza que la aplicación funcione ni en dispositivos móviles anteriores ni en posteriores que la marca Apple comercialice.

Tabla 68 – RH.01 Tipo de dispositivo.

Identificador	RH.02
Nombre	GPS en el dispositivo
Descripción	En los dispositivos que no esté el GPS integrado, no se puede asegurar una correcta localización.

Tabla 69 – RH.02 GPS en el dispositivo.

Identificador	RH.03
Nombre	Dimensión de la pantalla
Descripción	La aplicación está diseñada a tener unas interfaces cuyas medidas pueden ser: 1) 320x480 píxeles. 2) 640x960 píxeles. Si la dimensión de la pantalla del dispositivo móvil difiere de estas medidas, no se garantiza una correcta visualización de la aplicación.

Tabla 70 – RH.03 Dimensión de la pantalla.

Identificador	RH.04
Nombre	Conexión a internet
Descripción	La aplicación debe descargarse desde App Store, por lo que es necesario que el dispositivo móvil disponga de internet y cuenta en la tienda mencionada para poder descargarla.

Tabla 71 – RH.04 Conexión a internet.

5. Diseño

En este apartado se expone el diseño arquitectónico, la división en componentes y los diagramas de clase de la aplicación.

5.1. Modelo de Datos

Para llevar a cabo la gestión de los datos, se ha diseñado el modelo que se puede observar en la *imagen 22.* En esa imagen se puede ver cómo los niveles están relacionados con cada uno de los juegos. Para cada juego, se almacena en el modelo de datos la siguiente información:

- Juego Puzle: *Pista, Solución, Colocación inicial de las piezas del Puzle.*

- Juego Escítala: *Mensaje cifrado, Mensaje en claro, Tamaño de escitala.*

- Juego César: *Mensaje cifrado, Mensaje en claro, Desplazamiento.*

- Juego Playfair*: Mensaje cifrado, Clave a utilizar, Colocación de la clave, Mensaje en claro.*

- Juego Vigenere: *Mensaje cifrado, Mensaje en claro, Clave a utilizar.*

- Juego Diffie-Hellman: *Número p, Generador g, Clave pública del amigo, Mensaje cifrado, Clave compartida, Mensaje en claro.*

- Juego Pregunta/ Respuesta: Se divide a su vez en los siguientes juegos

    - Trivial: *Pregunta, Respuestas, Solución.*
    - Tabu: *Palabras, Solución.*
    - Pictionary: *Palabras, Imagen, Solución.*
    - Contador*: Pregunta, Respuestas.*

En el modelo de datos también se almacenan las puntuaciones de los juegos que se relacionan con sus respectivos juegos. Para cada juego se almacena la siguiente información: *Puntos Nivel, Número de intento, Superado/ No superado.*

Las opciones correspondientes al usuario se almacenan de igual modo en el modelo de datos, en este caso, los datos que se almacenan para las opciones son: *Puntos Totales, Alias, Nombre, Apellidos, Correo electrónico, Imagen*.

En la *Imagen 28*, los cuadros de color azul se corresponden con las entidades, mientras que los cuadros de color naranja se corresponden con los atributos de cada entidad. Las líneas azules representan las relaciones existentes entre las diferentes entidades del modelo.

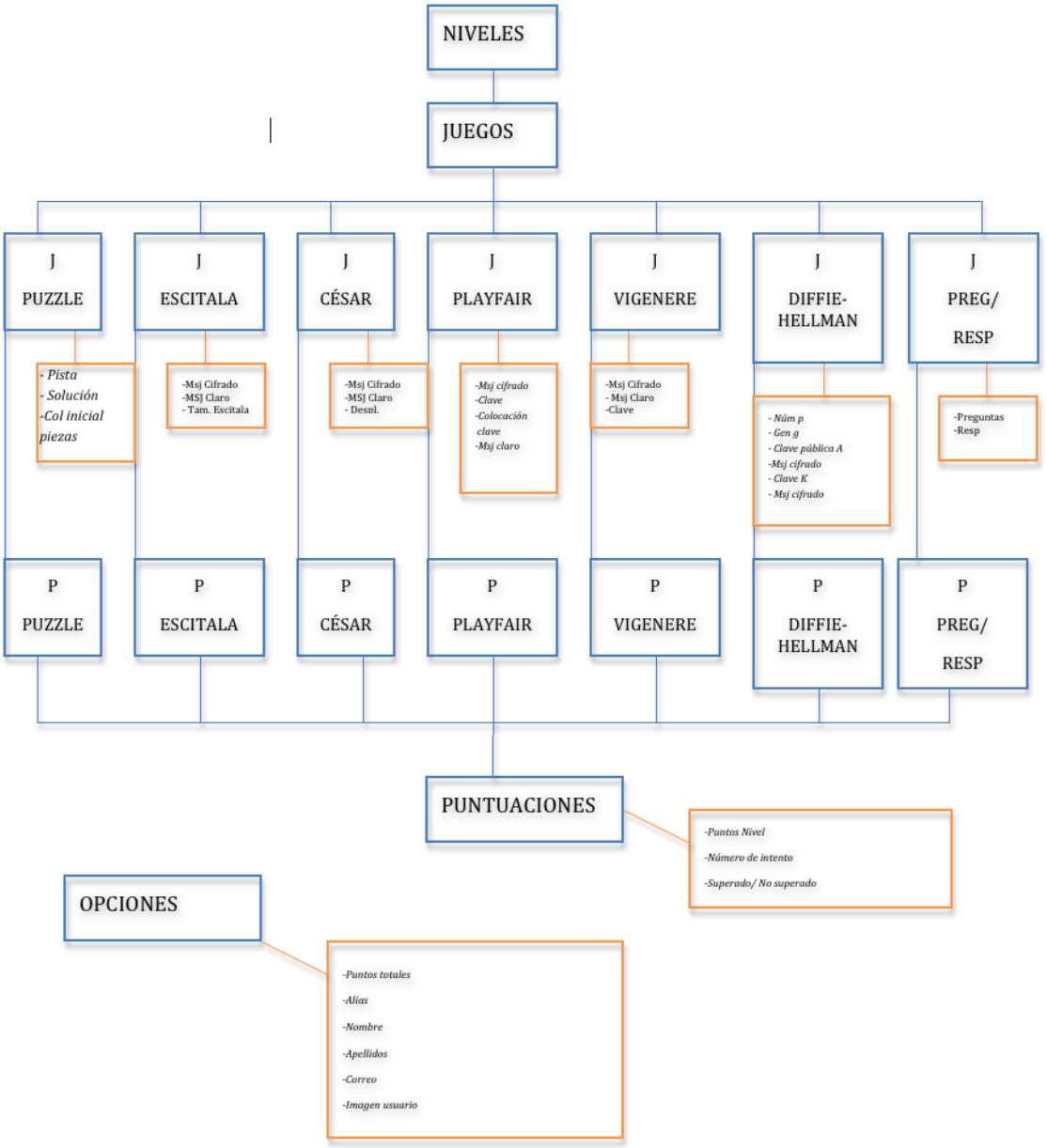


Imagen 28 – Modelo de datos.

5.2. Diseño de la Arquitectura del Sistema

Para la realización del diseño la arquitectura del sistema, se ha seguido la arquitectura modelo-vista-controlador (MVC). Esto se debe al propio *framework* de desarrollo de iPhone, que fomentan este tipo de arquitectura.

El MVC se divide en tres capas principales, cuyas funcionalidades están bien diferencias. Las funcionalidades de cada componente son las siguientes:

- **Modelo**: Representación específica de la información con la cual el sistema interactúa.
- **Vista:** Se encarga de la interacción con el usuario.
- **Controlador:** Representa la lógica mediante la cual se reciben peticiones del usuario, se transforman y se obtienen datos desde el modelo, para volver a transformarlos y devolverlos en un formato legible para el usuario.

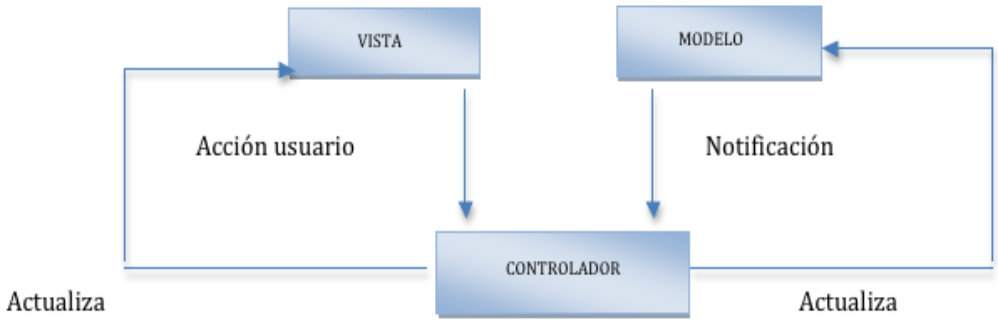


Imagen 29 – Modelo-Vista-Controlador.

5.2.1. Especificación de Componentes

En esta sección se realiza una especificación de componentes de manera detallada a través del diagrama de componentes que se muestra a continuación, así como mediante una explicación exhaustiva de cada componente.

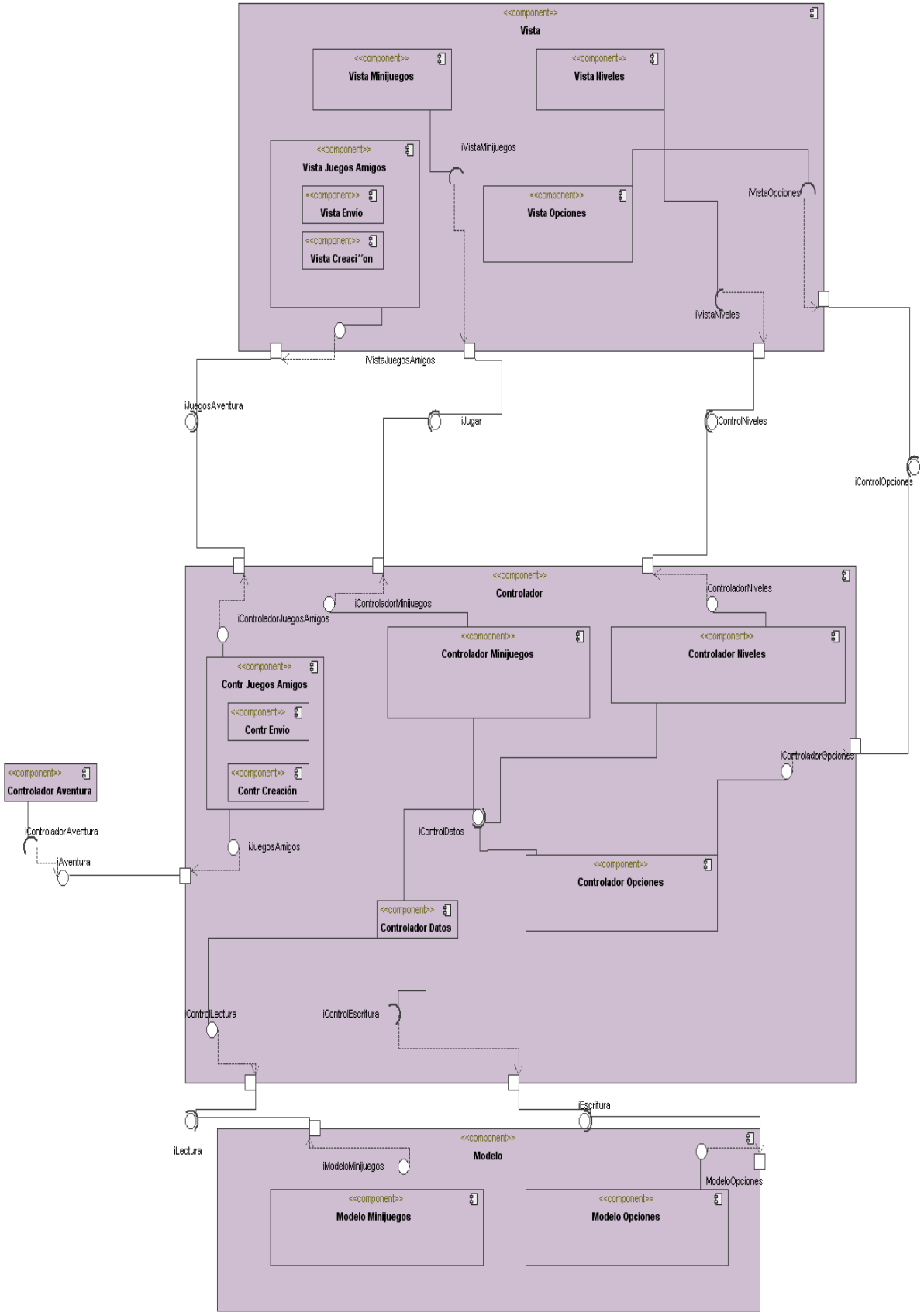


Imagen 30 – Diagrama de Componentes.

5.2.1.1. Capa Vista

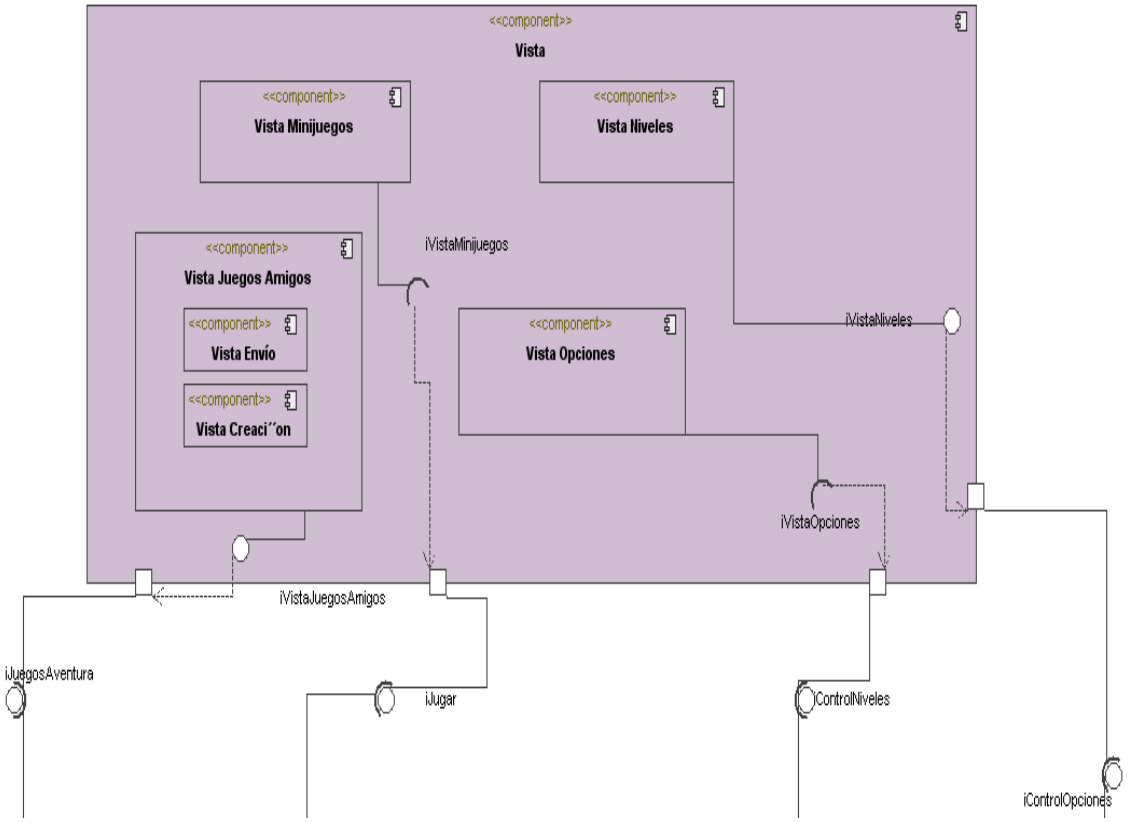


Imagen 31 – Capa Vista.

En la imagen anterior (*imagen 31)* se puede observar los componentes que conforman la capa de la vista. Este componente es el encargado de recoger las peticiones de usuario y pasárselas al componente *Controlador*. El componente *Controlador* devuelve la información necesaria y el componente *Vista*, muestra dicha información al usuario.

A continuación se detallan cada uno de los componentes internos del componente *Vista*:

- **Vista minijuegos**

Esta vista engloba todas las vistas relativas a cada uno de los juegos presentes en el sistema. La vista minijuegos requiere de la interfaz *iVistaMinijuegos,* para enviar peticiones a su correspondiente componente perteneciente a *Controlador*.

- **Vista niveles**

Esta vista engloba todas las vistas relativas a cada uno de los niveles de los juegos del sistema. La vista niveles requiere de la interfaz *iVistaNiveles* para enviar peticiones a su componente perteneciente a *Controlador*.

- **Vista opciones**

Esta vista engloba las vistas relativas a perfil y a resultados. La vista opciones requiere de la interfaz *iVistaOpciones* para enviar las peticiones de usuario a su componente perteneciente a *Controlador*.

- **Vista juegos amigos**

Esta vista contiene la vista de envío, y la vista de creación de juegos. Ambas vistas requieren de la interfaz iVistaJuegosAmigos para enviar peticiones de usuario a su componente perteneciente a *Controlador.*

5.2.1.2. Capa Controlador

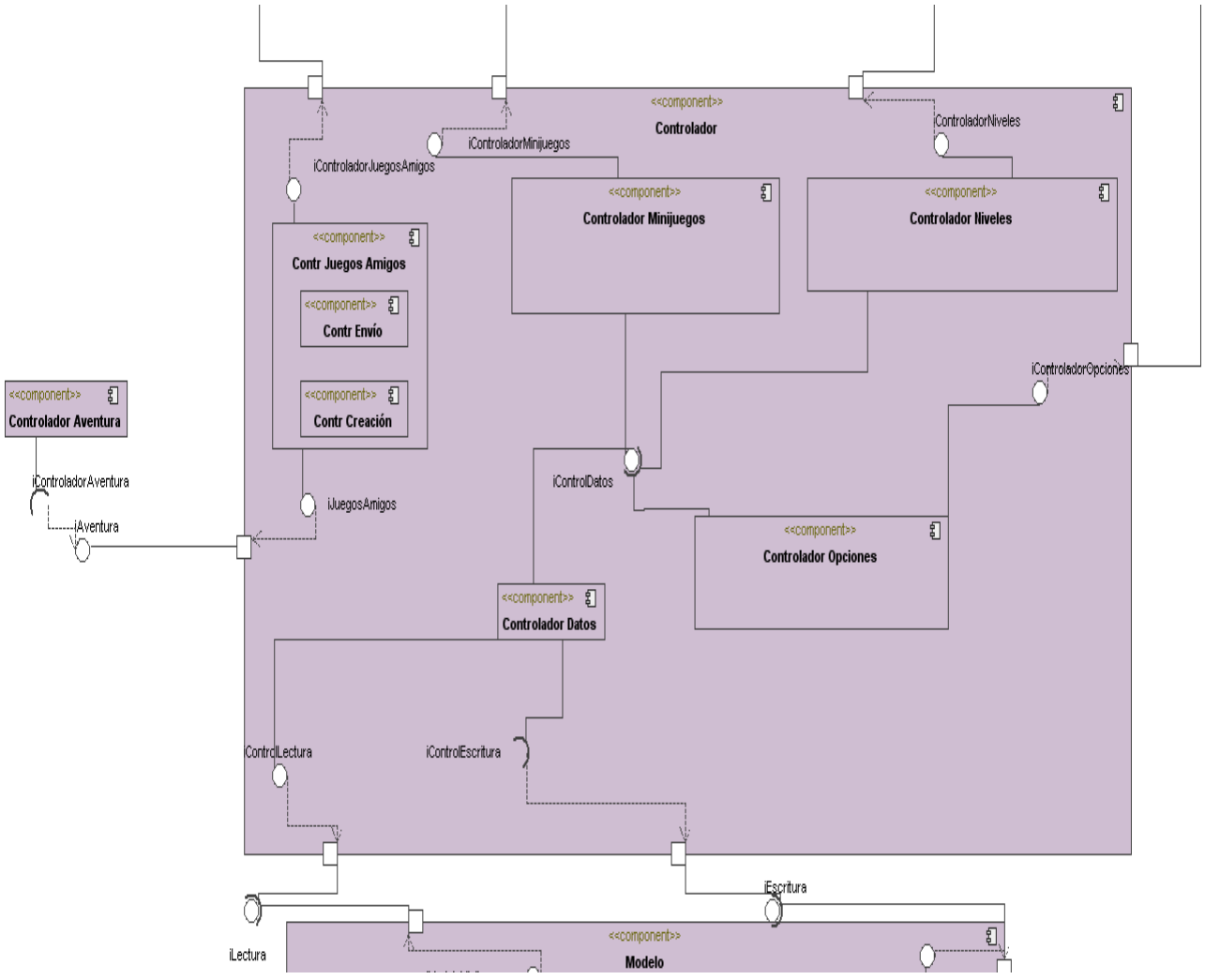


Imagen 32 – Capa Controlador.

En la imagen anterior (*Imagen 32*), se puede observar los componentes que forman parte de la capa del *Controlador*. Este componente contiene toda la lógica del sistema, el *Controlador* es el encargado de gestionar las peticiones de usuario que recibe del componente *Vista*, las procesa y hace a su vez la petición de datos al *Modelo*. El *Modelo* devuelve los datos pedidos, el *Controlador* los procesa y los prepara para devolverlos a la *Vista*.

Dentro de este componente se incluyen los siguientes controladores:

- **Controlador minijuegos**

Este controlador tiene la funcionalidad relativa a cada uno de los juegos del sistema, así como la gestión de sus correspondientes datos. Internamente está dividido en los controladores para cada uno de los juegos.

La interfaz que realiza es *iControladorMinijuegos* desde la cual recibe las peticiones de usuario procedentes de la vista y utiliza a su vez la interfaz *iControladorDatos* para leer y escribir los datos necesarios desde y hacia el *Modelo*.

- **Controlador niveles**

Este controlador tiene la funcionalidad relativa a cada uno de los niveles de los juegos del sistema, gestionando los datos relativos a cada uno de ellos. Internamente está dividido en los controladores para los niveles de cada uno de los juegos.

La interfaz que realiza es *iControladorNiveles* desde la cual recibe las peticiones de usuario procedentes de la vista y utiliza a su vez la interfaz *iControladorDatos* para leer y escribir los datos necesarios desde y hacia el *Modelo*.

- **Controlador opciones**

Este controlador tiene la funcionalidad y gestión de los datos relativos a las opciones del sistema. Internamente está dividido en los controladores relativos al perfil y a los resultados.

La interfaz que realiza es *iControladorOpciones* desde la cual recibe las peticiones de usuario procedentes de la vista y utiliza a su vez la interfaz *iControladorDatos* para leer y escribir los datos necesarios desde y hacia el Modelo.

- **Controlador juegos amigos**

Este controlador tiene la funcionalidad de comunicarse con el componente externo relativo al proyecto de *Aventura*. La gestión de las funcionalidades y de los datos no pertenecen al *Controlador*.

La interfaz que realiza es *iControladorJuegosAmigos* desde la cual recibe las peticiones de usuario procedentes de la vista y utiliza a su vez la interfaz

iJuegosAmigos para transferir el control de las funcionalidades de Juegos Amigos.

5.2.1.3. Capa Modelo

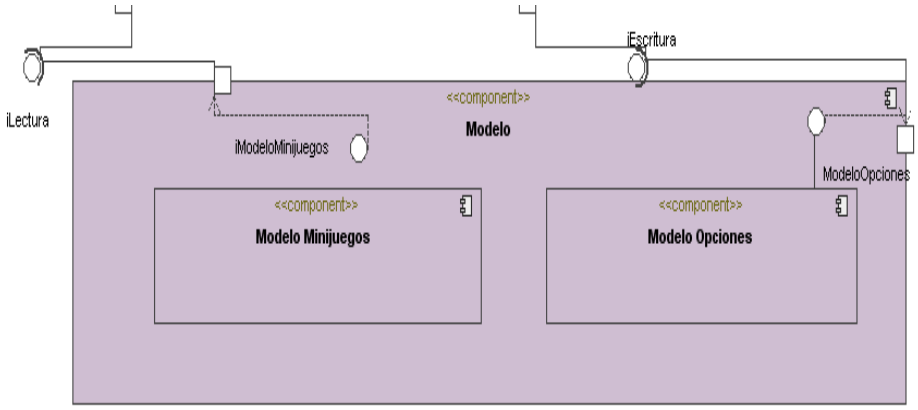


Imagen 33 – Capa Modelo.

- **Modelo minijuegos**

Este modelo almacena todos los datos referentes a los diferentes juegos del sistema. Utiliza la interfaz *iModeloMinijuegos* para comunicarse a su vez con las interfaces *iLectura* (para leer los datos almacenados en el modelo) e *iEscritura* (para escribir los nuevos datos procedentes del *Controlador*).

- **Modelo opciones**

Este modelo almacena todos los datos referentes a las opciones del sistema, el perfil y los resultados de los juegos. Utiliza la interfaz *iModeloOpciones* para comunicarse a su vez con las interfaces *iLectura* (para leer los datos almacenados en el modelo) e *iEscritura* (para escribir los nuevos datos procedentes del *Controlador*).

La estructura del modelo minijuegos y la del modelo opciones pueden consultarse en la sección *5.1 Modelo de Datos*.

5.3. Diseño Detallado del Sistema

5.3.1. Diseño de Clases

En la *Imagen 34*, se pueden observar las clases pertenecientes al componente de la *Vista.* Todas estas clases son las encargadas de mostrar los contenidos de una manera entendible por el usuario. Es por ello que todas estas clases no contienen métodos propios, ya que sólo se encargan de mostrar el contenido, pero sí tienen los métodos heredados de la clase *Vista.*

5.3.1.1. Diseño de Clases de la Vista

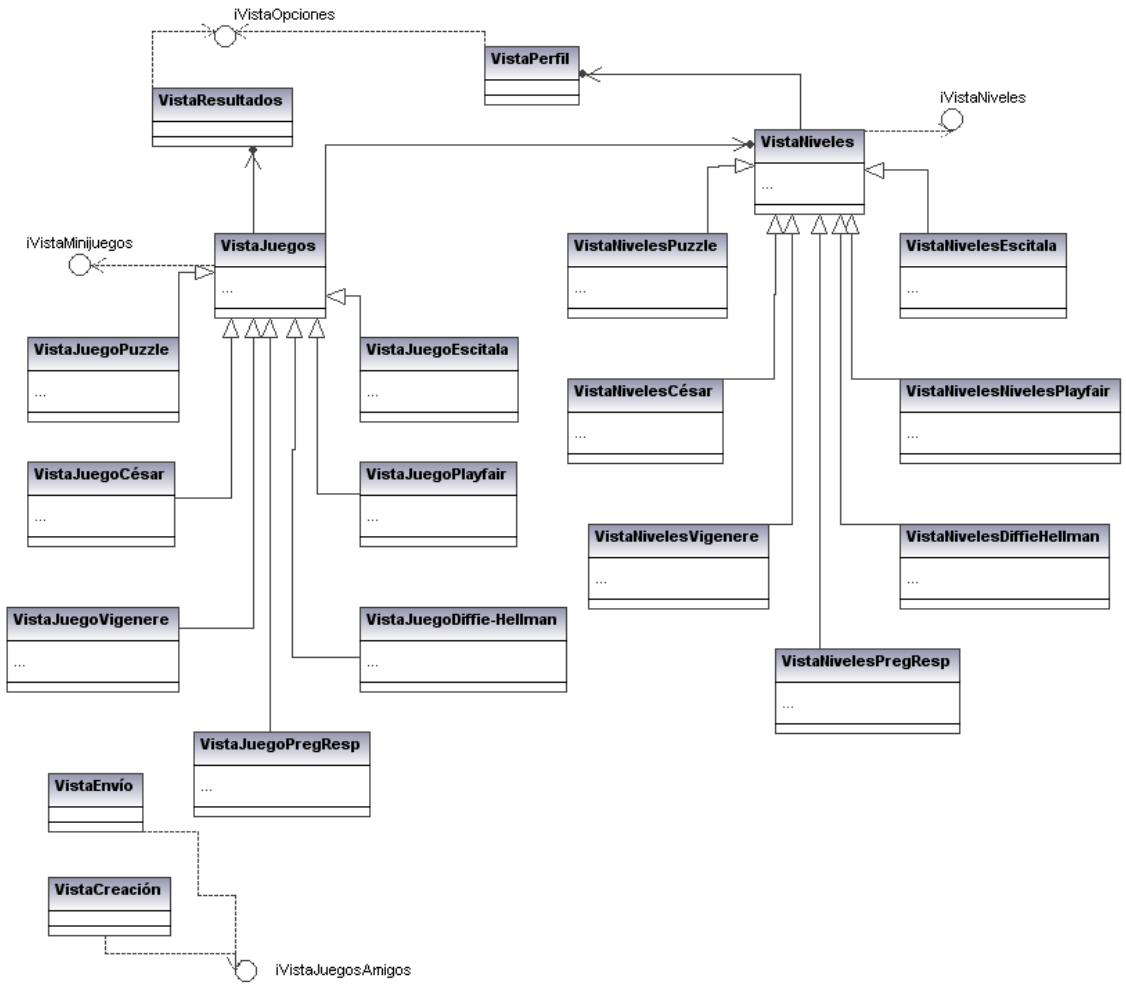


Imagen 34 – Clases del componente vista.

Partiendo de la clase *VistaJuegos*, el resto de clases para cada uno de los juegos, comparten elementos comunes. Esta clase *VistaJuegos*, se relaciona con la clase *VistaNiveles*, ya que dependiendo del nivel del juego, se mostrará unos elementos u otros. Para mostrar una información u otra en cada juego, la clase *VistaJuegos* debe relacionarse con *VistaResultados*.

VistaPerfil debe relacionarse con *VistaNiveles* para mostrar los niveles pertenecientes a un perfil en concreto. Por último, las vistas de *VistaEnvío* y *VistaCreación* no se relacionan con las vistas anteriores ya que forman parte de un proyecto diferente, *Aventura,* cuya funcionalidad se muestra en este proyecto a modo complementario.

5.3.1.1. Diseño de Clases del Controlador

A continuación se detallan las clases pertenecientes al C*ontrolador,* estas clases pueden verse en detalle en la *Imagen 35:*

Clase *Niveles*

Proporciona una interfaz común para el resto de niveles de juegos. En todas ellas se realiza la gestión necesaria para, dependiendo de las puntuaciones, superar los diferentes niveles, y en consecuencia, mostrar los datos referentes a cada nivel de juego.

Clase *Juegos*

Proporciona una interfaz común al resto de juegos que lo implementan. Cada uno de los juegos se encarga de la gestión de los datos de entrada necesarios para poder jugar, los datos necesarios de ayuda a la resolución del juego, la comprobación de la solución, así como la disposición de los elementos visuales en las interfaces correspondientes. Los juegos deben comunicarse con sus correspondientes niveles para hacer el cálculo de las puntuaciones, los niveles y los datos de cada nivel.

En los juegos donde sea necesaria la utilización de una clave, se hace indispensable la comunicación con otra clase que se encargue de la gestión de las claves. La utilización de esa clase se hace por parte de los juegos de Vigenere y Playfair.

Clase *Opciones*

Para la gestión de las opciones es necesaria la utilización de las clases para el perfil y para los resultados. En la clase Perfil se gestionan los datos referentes al usuario, así como sus elementos visuales. La funcionalidad de la clase Resultados se limita a la gestión visual de las puntuaciones de los niveles en cada juego.

Clase *gestorDatos*

La clase que gestiona los datos es gestorDatos, encargada de las interacciones tanto de lectura como de escritura de todos los datos referentes a los juegos, las puntuaciones, sus niveles, y las opciones con el modelo de datos. Esta clase es la que interactúa con el modelo, la encargada de leer los datos desde el modelo y escribir los datos que se necesita que persistan en el mismo.

Para ver un detalle de las clases que conforman el diagrama general de clases, se puede observar la *Imagen 35*:

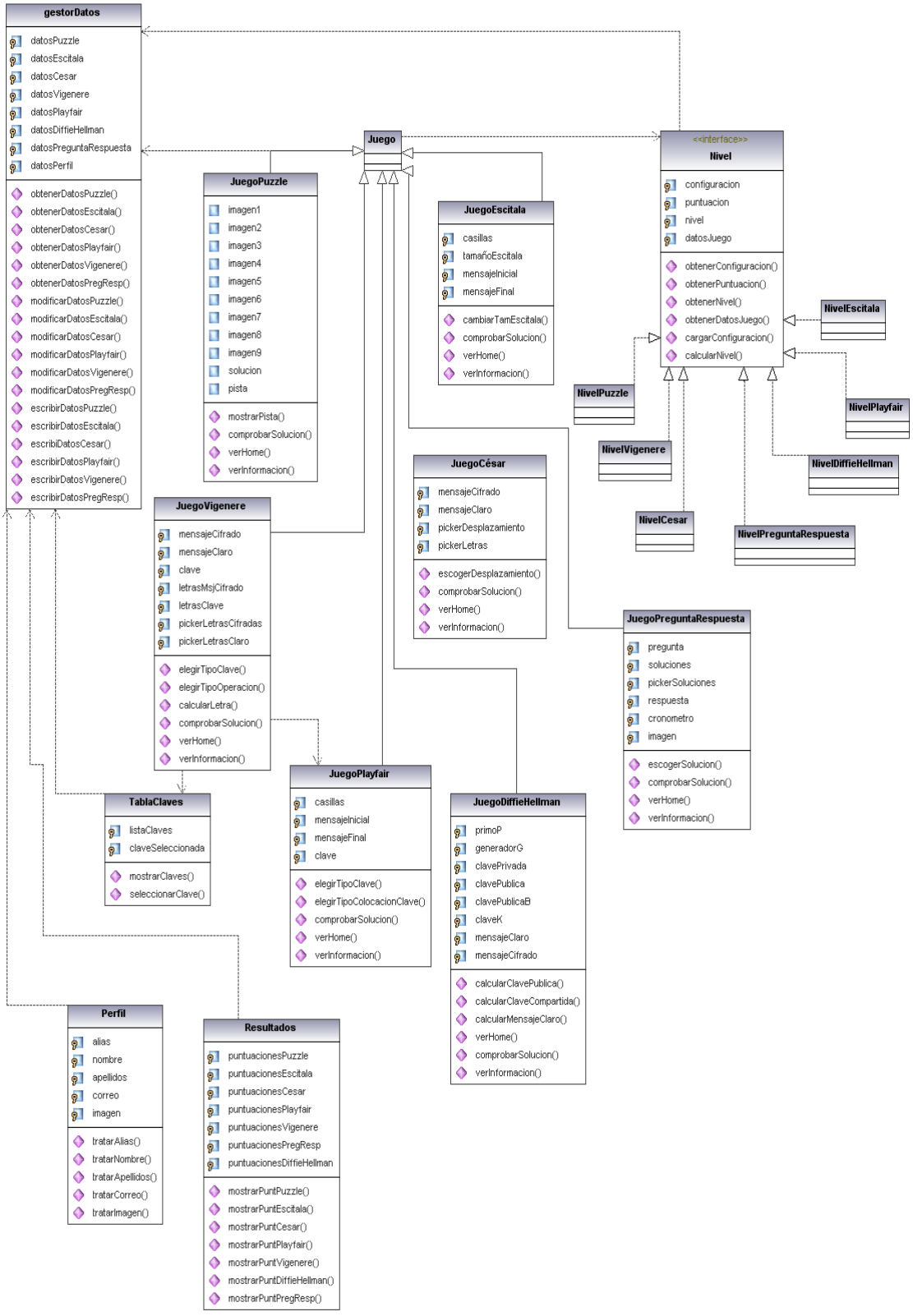


Imagen 35 – Diagrama de clases del *Controlador*.

5.4. Diagramas de Secuencia

Con los diagramas de secuencia se puede comprender el funcionamiento de la aplicación, y de este modo se puede contemplar el flujo normal de ejecución. A continuación se puede observar los diagramas de secuencia correspondiente a tres casos de uso que resumen la funcionalidad del sistema: *crear perfil*, *ver resultados juegos* y *jugar Puzle*. De los juegos posibles a los que se puede jugar, se ha indicado sólo uno, *jugar Puzle*, ya que todos los juegos siguen una dinámica similar. Con este diagrama de secuencia el lector puede hacerse una idea acerca del resto de interacciones de juegos.

5.4.1. Diagrama de Secuencia de Creación de Perfil

En este diagrama se puede observar la secuencia interna de las operaciones que hace el sistema cuando el usuario quiere crear un nuevo perfil.

El usuario introduce los datos necesarios y pulsa el botón de guardar. El sistema manda los datos del perfil a la clase *Perfil* y ésta a su vez se comunica con la clase *gestorDatos* que es la encargada de comunicarse con el modelo y mandarle los datos del perfil. Cuando finaliza la operación de guardado por parte del modelo, la clase *gestorDatos* devuelve a *Perfil* el resultado de la operación (normalmente será un mensaje de éxito, pero puede ocurrir algún imprevisto), y éste a su vez le manda un mensaje de éxito a la *vistaPerfil* para que el usuario lo vea.

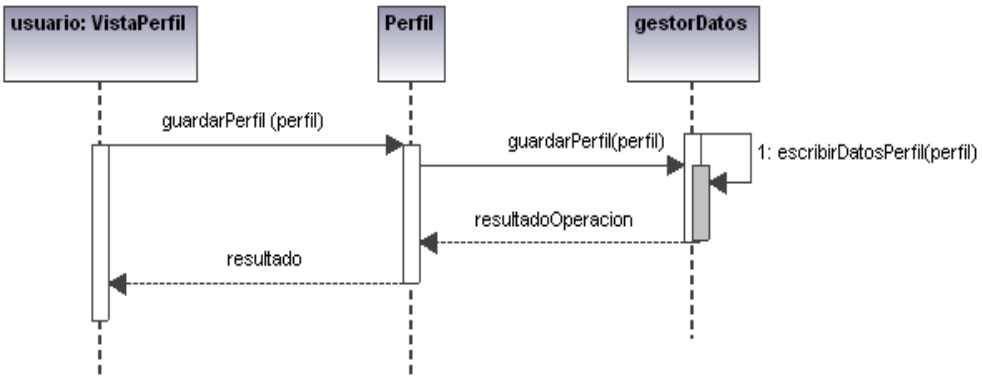


Imagen 36 – Diagrama de secuencia de creación de perfil.

5.4.1. Diagrama de Secuencia de Ver Resultados

En este diagrama se puede observar la secuencia interna de las operaciones que hace el sistema cuando el usuario quiere ver los resultados de los juegos.

El usuario navega hasta la interfaz de *vistaResultados* y pulsa el botón de ver resultados. El sistema manda la petición del usuario a la clase *Resultados.* Ésta debe comunicarse con la clase *GestorDatos* y debe pedirle todos y cada uno de los datos correspondientes a cada uno de los juegos. Para cada uno de los juegos el funcionamiento es el mismo: la clase *Resultados* pide la puntuación de un juego a *gestorDatos,* éste lo lee del modelo de datos y lo devuelve a la clase *Resultados* que a su vez lo muestra a la *vistaResultados* para satisfacer la petición del usuario.

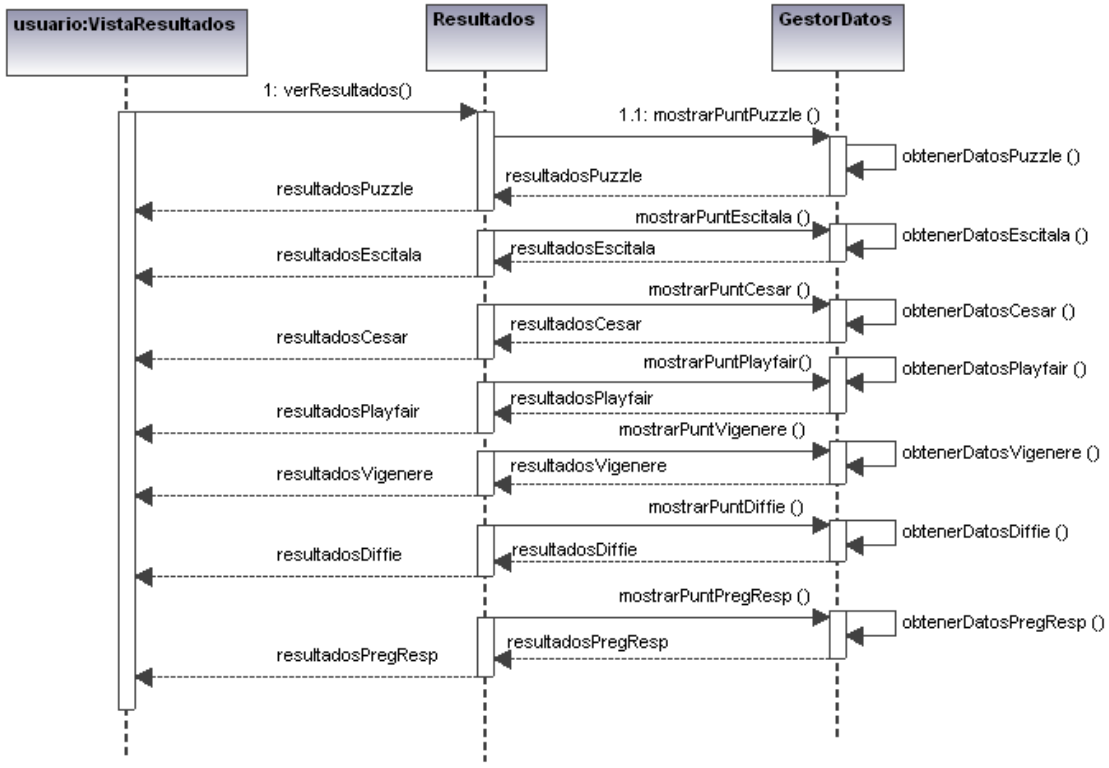


Imagen 37 – Diagrama de secuencia de ver resultados.

5.4.1. Diagrama de Jugar Puzle

En este diagrama se puede observar la secuencia interna de las operaciones que hace el sistema cuando el usuario quiere jugar al juego de Puzle.

El usuario a la interfaz de *vistaPuzle* y ésta hace una llamada a *NivelesPuzle*, ya que hay que averiguar los datos de qué nivel hay que cargar. *NivelesPuzle* pide a la clase *GestorDatos* que le devuelva el nivel en el que se encuentra el usuario en el juego de Puzle. *GestorDatos* lee del modelo los datos del juego de Puzle el nivel y lo devuelve a la clase *NivelesPuzle*. Como NivelesPuzle ya sabe el nivel, pide a *GestorDatos* los datos del Puzle correspondientes a ese nivel. Cuando se lo devuelve (después de leerlos del modelo), la clase *NivelesPuzle* carga la configuración del juego correspondiente con su nivel y a partir de ese momento el usuario puede jugar al Puzle.

El usuario en este momento puede querer que se muestre una pista, comprobar la solución del Puzle, comprobar la respuesta introducida, y siempre se hace directamente con *JuegoPuzle* sin necesidad de que ésta clase consulte al modelo ya que tiene previamente cargados los datos del juego.

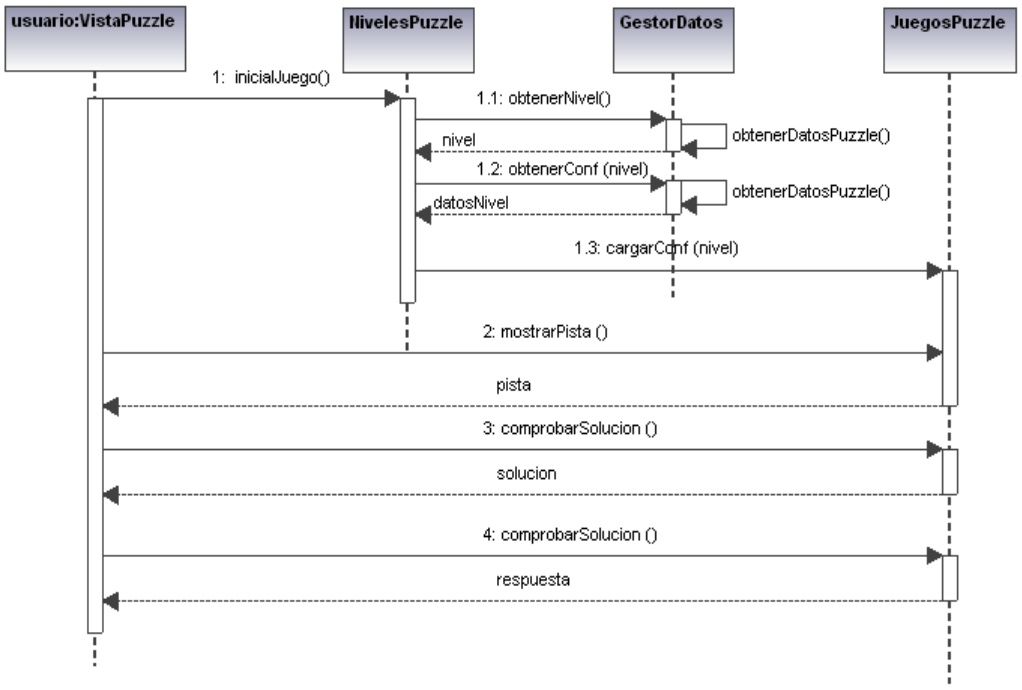


Imagen 38 – Diagrama de secuencia de jugar Puzle.

5.5. Interfaces

En esta sección se incluye un diagrama de navegación del sistema así como un diseño de las principales interfaces del sistema.

5.5.1. Diagrama de navegación del sistema

En la *Imagen 39* se muestra un diagrama de navegación del sistema. En el diagrama se pueden observar las diferentes pantallas que forman parte del sistema. Desde una pantalla se puede pasar a una o varias (se debe seguir las flechas que se muestran en la pantalla). Las cajas que aparecen en la imagen de color rojizo, son pantallas que aunque se puede acceder desde el proyecto de *Minijuegos*, no están asignados a él, forman parte del proyecto de *Aventura*, y desde aquí se proporciona un acceso directo al mismo.

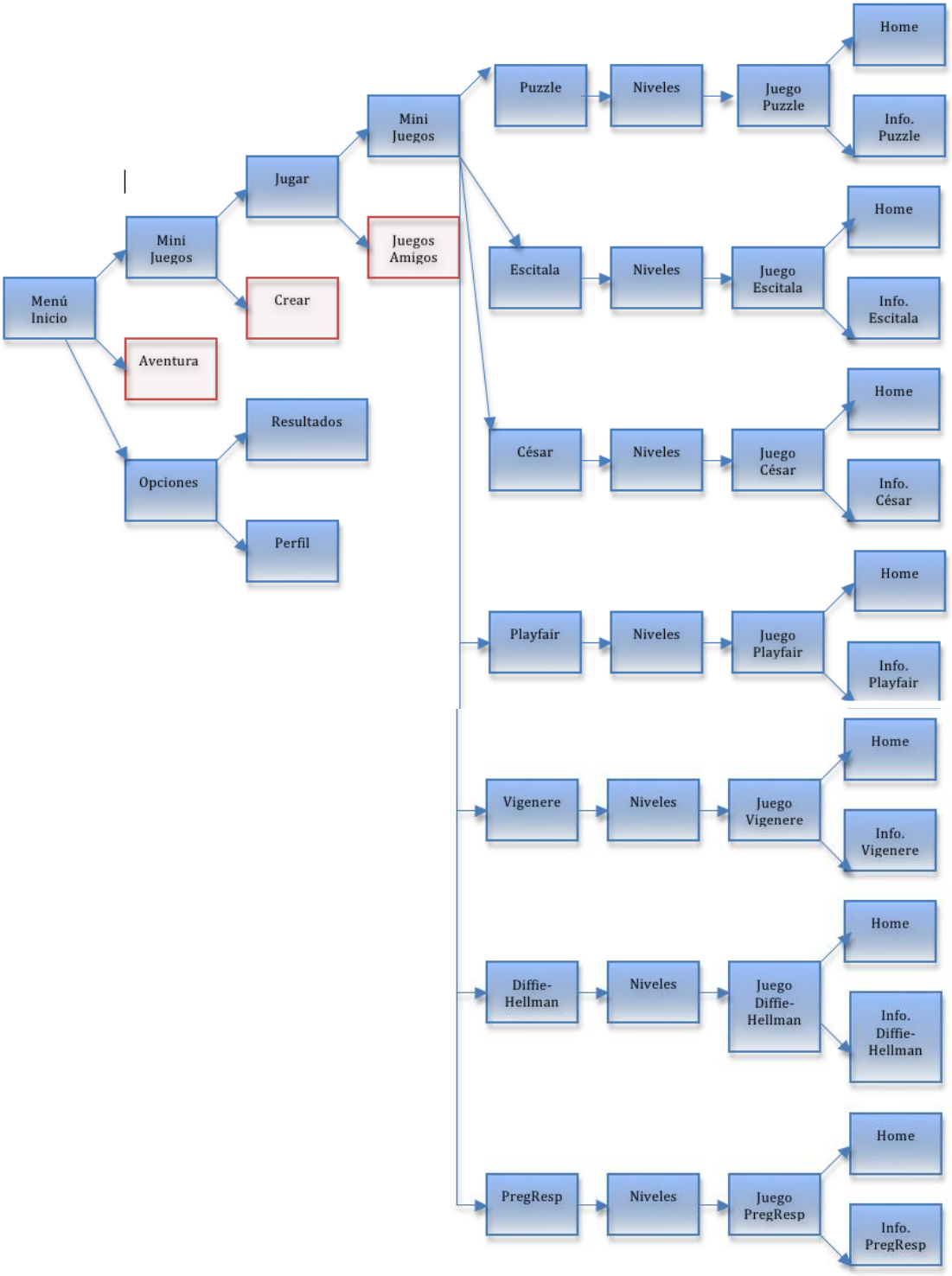


Imagen 39 – Diagrama de navegación del sistema.

5.5.2. Diseño de interfaces

En esta sección se muestra un prototipo de alto nivel del aspecto que deben presentar las interfaces del sistema. Se muestran las interfaces más genéricas, que engloban todos los tipos de interfaces que se pueden encontrar dentro de este proyecto, no obstante, cada interfaz en particular puede presentar las modificaciones oportunas que se ajusten a las necesidades propias. A continuación se muestran los distintos tipos de interfaces presentes en el sistema:

- **Interfaz Menú:** Tiene como objetivo proporcionar un menú de acceso a diferentes opciones que se presenten en el nivel donde se encuentra la interfaz. Tomando de ejemplo la interfaz de navegación principal mostrada en el diagrama de navegación (ver *Imagen 39*), su interfaz debe contener los elementos que se muestran en la *Imagen 40*.

- **Interfaz Juego:** Tiene como objetivo mostrar al usuario la pantalla donde se desarrolla el juego.

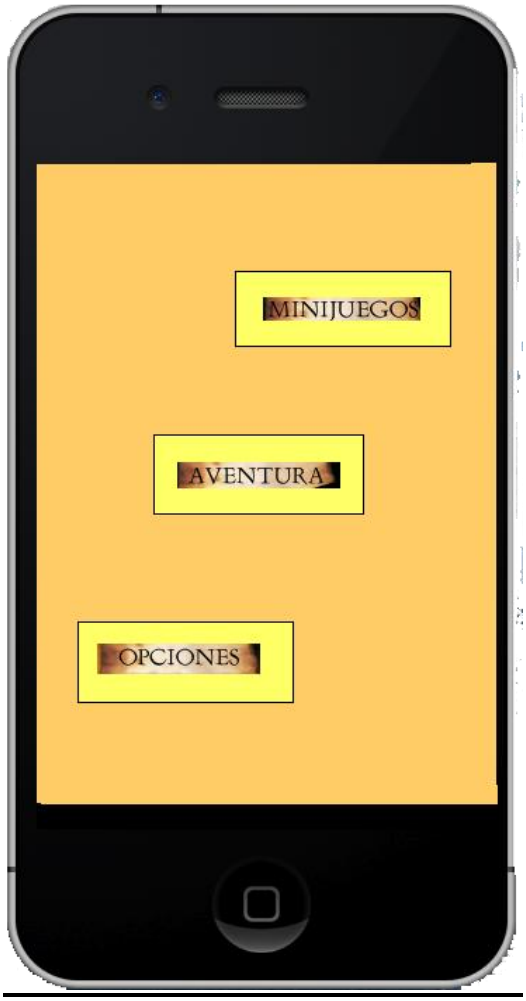


Imagen 40 – Interfaz Menú.

Imagen 41 – Interfaz Juego.

- **Interfaz Resultados:** Tiene como objetivo mostrar al usuario las puntuaciones conseguidas en los niveles de los juegos.

- **Interfaz Perfil:** Tiene como objetivo mostrar al usuario la pantalla donde poder añadir, modificar y borrar perfil de usuario.

Imagen 42 – Interfaz Resultados.

Imagen 43 – Interfaz Perfil.

6. Implementación y Pruebas

Una vez concluido el análisis y diseño, se pasa a la fase de implementación y pruebas partiendo del diseño arquitectónico y del diseño de componentes generados en la fase de diseño.

6.1. Implementación

6.1.1. Implementación de la arquitectura

La elección del lenguaje de programación no fue un problema a la hora de comenzar la implementación, ya que, debido a la imposición de desarrollar el sistema en iPhone, el lenguaje de programación inherente a la plataforma de desarrollo del SDK de iPhone es Objective-C.

Objective-C es un lenguaje de programación orientado a objetos. Está definido como una extensión al estándar del lenguaje de programación C. Siguiendo el estilo de C, proporciona funcionalidades muy potentes y ya previamente definidas que vienen incorporadas con el *framework* de programación.

Previamente al comienzo del proyecto de *Enigmatium-Minijuegos*, se desconocía el lenguaje de programación Objective-C, así como su plataforma de desarrollo. No obstante, debido a las asignaturas recibidas a lo largo de la carrera, se conocían lenguajes como Java o C. El conocimiento de Java, facilitó la comprensión de Objective-C al ser un lenguaje orientado a objetos, por lo que, el concepto de programación orientada a objetos no supuso problema alguno. De igual modo, el conocimiento previo de un lenguaje como C, supuso un avance en cuanto a la comprensión de la gestión de memoria.

Pese a conocer dos lenguajes de programación que en principio parecían muy similares a Objective-C, hubo que hacer un estudio previo a la implementación (el período de estudio del lenguaje puede apreciarse en la sección *3.2.1.2 Planificación real*). Durante este período se abarcaron temas de estudio como la comprensión del entorno de programación SDK proporcionado por Apple, funcionalidades específicas del lenguaje, *frameworks* y librerías disponibles para el desarrollo de esas funcionalidades específicas, particularidades y curiosidades del lenguaje, etc.

Las fuentes principales de información fueron internet, fundamentalmente la página de desarrollador de Apple, que proporciona una amplísima información tanto teórica como práctica [35]. Así mismo, se consultaron libros enfocados a la programación en Objective-C [36 y 37] y fundamentalmente [38].

Una vez superada la primera fase de documentación e implementación de ejemplos básicos, el desarrollo del sistema fue bastante fluido, incrementando la cantidad de líneas de código por jornada a medida que se avanzaba en la implementación.

Para el desarrollo del proyecto de *Minijuegos,* no se ha visto la necesidad de utilizar ninguna API externa en particular, únicamente la que viene por defecto, UIKit.

Para desarrollar del sistema, se empleó una metodología incremental iterativa (consultar sección *3 Gestión del Proyecto*), de manera que después de la implementación de cada bloque lógico funcional, se mantenía una reunión con el tutor para proponer posibles cambios o mejoras, o por el contrario, validar la iteración.

6.1.1. Implementación del modelo de datos

Para llevar a cabo la implementación del modelo de datos del sistema, se ha utilizado uno proporcionado por la propia plataforma de desarrollo de Apple, los ficheros *plist*.

Un fichero *plist* (*property list*) es un fichero de propiedades que contiene información de configuración. El contenido de estos ficheros es en *xml*, por lo que permite almacenamiento de datos *serializables* (enteros, booleanos, cadenas de texto, etc.), aunque lamentablemente no permite el almacenamiento de tipos de datos propios ni de imágenes, creándose la necesidad de serializar dichos datos.

Para el desarrollo de la presente aplicación, no se requiere la utilización de una base de datos, ya que la cantidad de datos que precisan de almacenamiento no son muy elevados. La gestión de los datos con ficheros *plist* es bastante asequible y no requiere un gran esfuerzo en el momento de la implementación.

Particularizando en el caso de la presente aplicación, los tipos de ficheros *plist* se pueden dividir en dos tipos:

- Ficheros de sólo lectura: Dentro de este grupo se encuentran los ficheros con los datos de los juegos.
- Ficheros de lectura y escritura: Dentro de este grupo se encuentran los ficheros que se modifican durante la ejecución del juego.

6.1.1.1. Ficheros de sólo lectura

Dentro de este grupo se encuentran los ficheros que contienen todos los datos necesarios para que la aplicación se ejecute de manera esperada. Los datos necesarios son todos los referentes a los juegos y a sus correspondientes niveles. A continuación se muestran los diferentes ficheros y su correspondiente estructura:

- *Puzle.plist*: Contiene los datos referentes a cinco juegos de Puzle.
    - *Nivel 1*
        - *Pista: String*
        - *Solución: String*
        - *Colocación inicial de las piezas del Puzle: Array*
    - *…*
    - *Nivel N*
        - *Pista: String*
        - *Solución: String*
        - *Colocación inicial de las piezas del Puzle: Array*

- *Escítala.plist:* Contiene los datos referentes a nueve juegos de Escítala.
    - *Nivel 1*
        - *Mensaje cifrado: String*
        - *Mensaje en claro: String*
        - *Tamaño de escitala: String*
    - *…*
    - *Nivel N*
        - *Mensaje cifrado: String*
        - *Mensaje en claro: String*
        - *Tamaño de escitala: String*
- *César.plist:* Contiene los datos referentes a ocho juegos de César.
    - *Nivel 1*
        - *Mensaje cifrado: String*
        - *Mensaje en claro: String*
        - *Desplazamiento: Integer*
    - *…*
    - *Nivel N*

        - *Mensaje cifrado: String*
        - *Mensaje en claro: String*
        - *Desplazamiento: Integer*

- *Playfair.plist:* Contiene los datos referentes a cinco juegos de Playfair.
    - *Nivel 1*
        - *Mensaje cifrado: String*
        - *Clave a utilizar: String*
        - *Colocación de la clave: Boolean*
        - *Mensaje en claro: String*
    - *…*
    - *Nivel N*
        - *Mensaje cifrado: String*
        - *Clave a utilizar: String*
        - *Colocación de la clave: Boolean*
        - *Mensaje en claro: String*
- *Vigenere.plist:* Contiene los datos referentes a seis juegos de Vigenere.
    - *Nivel 1*
        - *Mensaje cifrado: String*
        - *Mensaje en claro: String*
        - *Clave a utilizar: String*
    - *…*
    - *Nivel N*
        - *Mensaje cifrado: String*
        - *Mensaje en claro: String*
        - *Clave a utilizar: String*
- *Diffie-Hellman.plist:* Contiene los datos referentes a cuatro juegos de Diffie-Hellman.
    - *Nivel 1*
        - *Número p: Integer*
        - *Generador g: Integer*
        - *Clave pública del amigo: Integer*
        - *Mensaje cifrado: String*
        - *Clave compartida: Integer*
        - *Mensaje en claro: String*

- …
- *Nivel N*
  - *Número p: Integer*
  - *Generador g: Integer*
  - *Clave pública del amigo: Integer*
  - *Mensaje cifrado: String*
  - *Clave compartida: Integer*
  - *Mensaje en claro: String*

- *Trivial.plist:* Contiene los datos referentes a cinco juegos de Trivial.
  - *Nivel 1*
    - *Pregunta: String*
    - *Respuestas: Array de strings*
    - *Solución: Integer*
  - …
  - *Nivel N*
    - *Pregunta: String*
    - *Respuestas: Array de strings*
    - *Solución: Integer*
- *Tabu.plist:* Contiene los datos referentes a cinco juegos de Tabú.
  - *Nivel 1*
    - *Palabras: Array de strings*
    - *Solución: String*
  - …
  - *Nivel N*
    - *Palabras: Array de strings*
    - *Solución: String*
- *Pictionary.plist:* Contiene los datos referentes a cuatro juegos de Pictionary.
  - *Nivel 1*
    - *Palabras: Array de strings*
    - *Imagen: String con el nombre de la imagen*
    - *Solución: String*
  - …
  - *Nivel N*
    - *Palabras: Array de strings*

        - *Imagen: String con el nombre de la imagen*
        - *Solución: String*

- *contador.plist:* Contiene los datos referentes a cuatro juegos de contador.
    - *Nivel 1*
        - *Pregunta: String*
        - *Respuestas: Array de strings*
    - *…*
    - *Nivel N*
        - *Pregunta: String*
        - *Respuestas: Array de strings*

6.1.1.2. Ficheros de lectura y escritura

Dentro de este grupo se encuentran los ficheros que van a modificar su contenido a medida que el usuario interactúa con la aplicación. Al comienzo de esta interacción, estos ficheros se encuentran vacíos y, conforme el usuario puntúa en los niveles de los juegos, los niveles y las puntuaciones se irán modificando dentro de estos ficheros *plist*. Los ficheros *plist* pertenecientes a este grupo y sus estructuras internas se muestran a continuación:

- *Opciones.plist*: Contiene la información referente al perfil del usuario.
    - *Puntos totales: Integer*
    - *Alias: String*
    - *Nombre: String*
    - *Apellidos: String*
    - *Correo: String*
    - *Imagen usuario: String con el nombre de la imagen*
- Puntuaciones*Juego*.plist: Donde *Juego* se corresponde con cada uno de los juegos presentes en el sistema. Así, los plist de puntuaciones son: *puntuacionesEscitala.plist*, *puntuacionesPuzle.plist*, *puntuacionesCésar.plist*, *puntuacionesPlayfair.plist*, *puntuacionesVigenere.plist*, *puntuacionesDiffieHellman.plist* y *puntuacionesPregResp.plist.*

  La estructura de cada uno de estos plist es idéntica, ya que la gestión de la información debe ser la misma para cada juego, con independencia de los datos. La estructura de cada uno de estos ficheros es:
    - *Último nivel superado: Integer*
    - *Nivel 1*
        - *Puntos Nivel: Integer*

        - *Número de intento: Integer*
        - *Superado/ No superado: Boolean*
    - *Nivel 2*
        - *Puntos Nivel: Integer*
        - *Número de intento: Integer*
        - *Superado/ No superado: Boolean*
    - *…*
    - *…*
    - *Nivel N*
        - *Puntos Nivel: Integer*
        - *Número de intento: Integer*
        - *Superado/ No superado: Boolean*

6.2. Pruebas

Para comprobar la correcta funcionalidad del sistema, se han realizado pruebas a lo largo de cada una de las iteraciones del sistema. De este modo se ha podido observar con detalle el correcto funcionamiento del sistema.

6.2.1. Pruebas de Aceptación

Las pruebas de aceptación realizadas al sistema en las sucesivas iteraciones se muestran a continuación en forma de tablas.

Identificador	PA-XXX
Descripción	
Resultado	

Tabla 72 – Plantilla de prueba de aceptación.

- **Identificador**: Identificar único, donde *PA* indica *pruebas de aceptación* y XXX indica un número de 3 dígitos, comenzando por el 001 finalizando en el 999 de manera incremental consecutiva.

- **Descripción**: Incluye una descripción sobre la prueba de aceptación realizada.

- **Resultado**: Resultado de la prueba de aceptación, cuyo resultado podrá ser: *Superada/No superada*.

Identificador	PA-001
Descripción	Se comprueba el acceso a la pantalla de resultados, así como la obtención de los puntos en cada nivel de los juegos. Los puntos deben ser correctos.
Resultado	*Superada*

Tabla 73 – Prueba de Aceptación PA-001.

Identificador	PA-002
Descripción	Se comprueba el acceso a la pantalla de perfil. Se introducen los datos relativos al perfil (alias, nombre, apellidos, correo electrónico e imagen). Se guardan los datos. El sistema debe mostrar un mensaje indicando que los datos se guardan correctamente.
Resultado	*Superada*

Tabla 74 – Prueba de Aceptación PA-002.

Identificador	PA-003
Descripción	Se comprueba el acceso a la pantalla de perfil. Se modifican los datos relativos al perfil (alias, nombre, apellidos, correo electrónico e imagen). Se guardan los datos. El sistema debe mostrar un mensaje indicando que los datos se modifican correctamente.
Resultado	*Superada*

Tabla 75 – Prueba de Aceptación PA-003.

Identificador	PA-004
Descripción	Se comprueba el acceso a la pantalla de perfil. Se elimina el perfil por completo. El sistema debe mostrar un mensaje indicando que los datos se eliminan correctamente.
Resultado	*Superada*

Tabla 76 – Prueba de Aceptación PA-004.

Identificador	PA-005

Descripción	Se comprueba el acceso a la pantalla del juego de Puzle. Se siguen los pasos necesarios para superar el juego. El sistema debe indicar al usuario que ha superado el juego. El sistema no debe mostrar ningún mensaje de error relativo a un fallo en la programación.
Resultado	*Superada*

Tabla 77 – Prueba de Aceptación PA-005.

Identificador	PA-006
Descripción	Se comprueba el acceso a la pantalla del juego de Puzle. Se comprueba la solución introduciendo datos aleatorios para fallar el juego. El sistema debe indicar al usuario que la solución introducida no es correcta. El sistema no debe mostrar ningún mensaje de error relativo a un fallo en la programación.
Resultado	*Superada*

Tabla 78 – Prueba de Aceptación PA-006.

Identificador	PA-007
Descripción	Se comprueba el acceso a la pantalla del juego de Escítala. Se siguen los pasos necesarios para superar el juego. El sistema debe indicar al usuario que ha superado el juego. El sistema no debe mostrar ningún mensaje de error relativo a un fallo en la programación.
Resultado	*Superada*

Tabla 79 – Prueba de Aceptación PA-007.

Identificador	PA-008
Descripción	Se comprueba el acceso a la pantalla del juego de Escítala. Se comprueba la solución introduciendo datos aleatorios para fallar el juego. El sistema debe indicar al usuario que la solución introducida no es correcta. El sistema no debe mostrar ningún mensaje de error relativo a un fallo en la programación.
Resultado	*Superada*

Tabla 80 – Prueba de Aceptación PA-008.

Identificador	PA-009
Descripción	Se comprueba el acceso a la pantalla del juego de Vigenere. Se siguen los pasos necesarios para superar el juego. El sistema debe indicar al usuario que ha superado el juego. El sistema no debe mostrar ningún mensaje de error relativo a un fallo en la programación.
Resultado	*Superada*

Tabla 81 – Prueba de Aceptación PA-009.

Identificador	PA-010
Descripción	Se comprueba el acceso a la pantalla del juego de Vigenere. Se comprueba la solución introduciendo datos aleatorios para fallar el juego. El sistema debe indicar al usuario que la solución introducida no es correcta. El sistema no debe mostrar ningún mensaje de error relativo a un fallo en la programación.
Resultado	*Superada*

Tabla 82 – Prueba de Aceptación PA-010.

Identificador	PA-011
Descripción	Se comprueba el acceso a la pantalla del juego de Playfair. Se siguen los pasos necesarios para superar el juego. El sistema debe indicar al usuario que ha superado el juego. El sistema no debe mostrar ningún mensaje de error relativo a un fallo en la programación.
Resultado	*Superada*

Tabla 83 – Prueba de Aceptación PA-011.

Identificador	PA-012
Descripción	Se comprueba el acceso a la pantalla del juego de Playfair. Se comprueba la solución introduciendo datos aleatorios para fallar el juego. El sistema debe indicar al usuario que la solución introducida no es correcta. El sistema no debe mostrar ningún mensaje de error relativo a un fallo en la programación.
Resultado	*Superada*

Tabla 84 – Prueba de Aceptación PA-012.

Identificador	PA-013
Descripción	Se comprueba el acceso a la pantalla del juego de César. Se siguen los pasos necesarios para superar el juego. El sistema debe indicar al usuario que ha superado el juego. El sistema no debe mostrar ningún mensaje de error relativo a un fallo en la programación.
Resultado	*Superada*

Tabla 85 – Prueba de Aceptación PA-013.

Identificador	PA-014
Descripción	Se comprueba el acceso a la pantalla del juego de César. Se comprueba la solución introduciendo datos aleatorios para fallar el juego. El sistema debe indicar al usuario que la solución introducida no es correcta. El sistema no debe mostrar ningún mensaje de error relativo a un fallo en la programación.
Resultado	*Superada*

Tabla 86 – Prueba de Aceptación PA-014.

Identificador	PA-015
Descripción	Se comprueba el acceso a la pantalla del juego de Diffie-Hellman. Se siguen los pasos necesarios para superar el juego. El sistema debe indicar al usuario que ha superado el juego. El sistema no debe mostrar ningún mensaje de error relativo a un fallo en la programación.
Resultado	*Superada*

Tabla 87 – Prueba de Aceptación PA-015.

Identificador	PA-016
Descripción	Se comprueba el acceso a la pantalla del juego de Diffie-Hellman. Se comprueba la solución introduciendo datos aleatorios para fallar el juego. El sistema debe indicar al usuario que la solución introducida no es correcta. El sistema no debe mostrar ningún mensaje de error relativo a un fallo en la programación.

Resultado	*Superada*

Tabla 88 – Prueba de Aceptación PA-016.

Identificador	PA-017
Descripción	Se comprueba el acceso a la pantalla del juego de Pregunta/Respuesta. Se siguen los pasos necesarios para superar el juego. El sistema debe indicar al usuario que ha superado el juego. El sistema no debe mostrar ningún mensaje de error relativo a un fallo en la programación.
Resultado	*Superada*

Tabla 89 – Prueba de Aceptación PA-017.

Identificador	PA-018
Descripción	Se comprueba el acceso a la pantalla del juego de Pregunta/Respuesta. Se comprueba la solución introduciendo datos aleatorios para fallar el juego. El sistema debe indicar al usuario que la solución introducida no es correcta. El sistema no debe mostrar ningún mensaje de error relativo a un fallo en la programación.
Resultado	*Superada*

Tabla 90 – Prueba de Aceptación PA-018.

Identificador	PA-019
Descripción	Se comprueba el acceso a la pantalla de envío de juegos. El sistema debe redirigir al usuario al módulo de *Aventura*.
Resultado	*Superada*

Tabla 91 – Prueba de Aceptación PA-019.

Identificador	PA-020
Descripción	Se comprueba el acceso a la pantalla de envío de creación de juegos. El sistema debe redirigir al usuario al módulo de *Aventura*.
Resultado	*Superada*

Tabla 92 – Prueba de Aceptación PA-020.

Identificador	PA-021
Descripción	Se comprueba el acceso a la pantalla de menú inicio desde cualquier punto de la aplicación. El sistema no debe mostrar ningún mensaje de error relativo a un fallo en la programación.
Resultado	*Superada*

Tabla 93 – Prueba de Aceptación PA-021.

Identificador	PA-022
Descripción	Se comprueba el acceso a la pantalla de menú de minijuegos desde cualquier punto de la aplicación. El sistema no debe mostrar ningún mensaje de error relativo a un fallo en la programación.
Resultado	*Superada*

Tabla 94 – Prueba de Aceptación PA-022.

Identificador	PA-023
Descripción	Se comprueba el acceso a la pantalla del juego de Puzle. Desde esta pantalla, se comprueba el acceso a la información del juego. El sistema no debe mostrar ningún mensaje de error relativo a un fallo en la programación.
Resultado	*Superada*

Tabla 95 – Prueba de Aceptación PA-023.

Identificador	PA-024
Descripción	Se comprueba el acceso a la pantalla del juego de Escítala. Desde esta pantalla, se comprueba el acceso a la información del juego. El sistema no debe mostrar ningún mensaje de error relativo a un fallo en la programación.
Resultado	*Superada*

Tabla 96 – Prueba de Aceptación PA-024.

Identificador	PA-025
Descripción	Se comprueba el acceso a la pantalla del juego de Vigenere. Desde esta pantalla, se comprueba el acceso a la información del juego. El sistema no debe mostrar ningún mensaje de error relativo a un fallo en la programación.
Resultado	*Superada*

Tabla 97 – Prueba de Aceptación PA-025.

Identificador	PA-026
Descripción	Se comprueba el acceso a la pantalla del juego de Playfair. Desde esta pantalla, se comprueba el acceso a la información del juego. El sistema no debe mostrar ningún mensaje de error relativo a un fallo en la programación.
Resultado	*Superada*

Tabla 98 – Prueba de Aceptación PA-026.

Identificador	PA-027
Descripción	Se comprueba el acceso a la pantalla del juego de César. Desde esta pantalla, se comprueba el acceso a la información del juego. El sistema no debe mostrar ningún mensaje de error relativo a un fallo en la programación.
Resultado	*Superada*

Tabla 99 – Prueba de Aceptación PA-027.

Identificador	PA-028
Descripción	Se comprueba el acceso a la pantalla del juego de Diffie-Hellman. Desde esta pantalla, se comprueba el acceso a la información del juego. El sistema no debe mostrar ningún mensaje de error relativo a un fallo en la programación.
Resultado	*Superada*

Tabla 100 – Prueba de Aceptación PA-028.

Identificador	PA-029
Descripción	Se comprueba el acceso a la pantalla del juego de Pregunta/Respuesta. Desde esta pantalla, se comprueba el acceso a la información del juego. El sistema no debe mostrar ningún mensaje de error relativo a un fallo en la programación.
Resultado	*Superada*

Tabla 101 – Prueba de Aceptación PA-029.

6.2.2. Matriz de Trazabilidad: RS – PA

Para poder visualizar fácil y rápidamente las relaciones entre requisitos de software y pruebas de aceptación, se ha creado la matriz de trazabilidad que se muestra en la *Tabla 102,* donde se puede apreciar cómo con las pruebas de aceptación se han cubierto todos los requisitos de software.

	PA.001	PA.002	PA.003	PA.004	PA.005	PA.006	PA.007	PA.008	PA.009	PA.010	PA.011	PA.012	PA.013	PA.014	PA.015	PA.016	PA.017	PA.018	PA.019	PA.020	PA.021	PA.022	PA.023	PA.024	PA.025	PA.026	PA.027	PA.028	PA.029
RS.F.001.01													X	X															
RS.F.001.02									X	X																			
RS.F.001.03											X	X																	
RS.F.001.04							X	X																					
RS.F.001.05															X	X													
RS.F.001.06					X	X																							
RS.F.001.07																	X	X											
RS.F.003.01					X		X		X		X		X		X		X												
RS.F.003.02					X		X		X		X		X		X		X												
RS.F.004.01																							X	X	X	X	X	X	X
RS.F.006.01																			X	X	X								
RS.F.006.02																						X							
RS.F.007.01	X																												
RS.F.008.01		X	X	X																									

Tabla 102 – Matriz de trazabilidad.

6.2.3. Pruebas de Usabilidad

Finalizada la de *Minijuegos*, se realizó un test de usabilidad a diferentes usuarios, tanto antiguos estudiantes de la asignatura de *Seguridad en las Tecnologías de la Información*, como estudiantes actuales de la misma. De los usuarios encuestados, algunos son usuarios habituales de iPhone y el resto utilizan otras plataformas móviles.

Las pruebas de usabilidad realizadas consisten en proporcionar al usuario un iPhone o iPad con la aplicación instalada. Se le pide al usuario que interactúe con ella y que juegue al menos una vez a cada uno de los juegos. Esta prueba dura alrededor de una hora.

Después de una hora de observación al usuario, se observan comportamientos que adopta con la aplicación. Algunos de estos comportamientos son repetidos por varios usuarios, por lo que se decide realizar pequeñas modificaciones en algunas interfaces de los juegos.

Al finalizar la prueba, se les pasa un test de valoración del sistema *(*Anexo 2. Test de evaluación por parte del usuario final)

Con un total de 6 evaluaciones, la clasificación del perfil de los usuarios de realiza de la siguiente manera:

	Usuario iPhone	Usuario otras plataformas
Estudiante Actual	1	1
Antiguo Estudiante	0	4

Tabla 103 – Clasificación perfil usuarios.

Realizando una media de las encuestas realizadas, se obtienen los siguientes resultados valorados sobre 10 puntos:

- Intuitividad del sistema: **7**

  La mayor parte de los usuarios esperan que las letras se coloquen en las casillas de manera automática, sin tener que realizar ellos ningún esfuerzo.

- Correcta interpretación de los botones de las interfaces: **8**

  La mayor parte de los usuarios comprenden a simple vista la finalidad de cada botón.

- Utilidad de la ayuda: **9,5**

Pese a que los usuarios no muestran una predisposición natural a utilizar la funcionalidad de ayuda, finalmente acaban consultándola, obteniendo un alto grado de satisfacción por la información obtenida.

- Intuitividad de la navegación: **7,5**

  Los usuarios, en su gran mayoría, no se sienten perdidos en ningún punto de la aplicación y son capaces de retroceder o continuar hacia otras funcionalidades sin ningún problema.

- Utilidad de la aplicación para el refuerzo de los conocimientos en algoritmos de contenido criptográfico: **9,8**

  Los usuarios encuestados encuentran de mucha utilidad la aplicación en cuanto a su contenido docente. Usuarios que han superado la asignatura "*Seguridad en las Tecnologías de la Información*", recuerdan los algoritmos y los ponen en práctica.

- Intención de compra de la aplicación: **Sí** (4 personas), **No** (2 personas).

  De los usuarios consultados, 4 de ellos se muestran dispuestos a comprar la aplicación en caso de ponerse a la venta, mientras que 2 de ellos se niegan a tener que pagar por un recurso docente. Consideran que la universidad debería proporcionarlo gratuitamente.

- Valoración general: **7,8**

  Para concluir, los usuarios valoran la aplicación con una media de 7,8.

Cabe destacar que los usuarios que mejor supieron utilizar la aplicación fueron los dos que se encontraban cursando la asignatura de "*Seguridad en las Tecnologías de la Información*".

7. Conclusiones y líneas futuras

En esta sección se ofrecen las conclusiones obtenidas una vez finalizado el proyecto y las líneas futuras por las que sería conveniente continuar el proyecto.

7.1. Conclusiones

El aprendizaje de los algoritmos criptográficos es un proceso costoso debido a su elevado contenido matemático, por lo que requiere de ayudas complementarias durante su estudio. Con el presente proyecto se cubre esa necesidad de complemento al estudio mediante juegos dinámicos y entretenidos sobre una plataforma móvil.

Los resultados que se pueden obtener mediante el uso de esta aplicación pueden llegar a ser muy satisfactorios, ya se puede intuir este resultado a través de las pruebas de usabilidad realizadas (ver *6.2.3Pruebas de Usabilidad*), en las cuales los usuarios que se encontraban preparando el temario de la asignatura respondían con mucha agilidad a los juegos propuestos.

Al contener una gran variedad de juegos con niveles dentro de cada uno, el usuario puede adecuar el nivel personal de conocimientos con los niveles presentes en la aplicación. Así mismo, con todos aquellos conceptos con los que el usuario no se sienta familiarizado, puede consultarlos dentro de la propia información, por lo que también se puede encontrar dentro de ella contenido teórico acerca de algoritmos criptográficos.

Al integrar el proyecto de *Minijuegos* con el proyecto de *Aventura,* en la aplicación común llamada *Enigmatium*, se dota a la aplicación conjunta de un contenido no sólo docente sino también divertido (consultar el proyecto *Aventura*). A través de *Enigmatium* el usuario podrá jugar a los juegos propuestos, crear nuevos juegos para que otros usuarios puedan utilizarlos o seguir una aventura con pistas repartidas en la ciudad de Leganés.

En cuanto a la implementación de la aplicación, al seguir una arquitectura basada en el modelo vista controlador, se ha dotado al sistema de una modularidad buena para que en un futuro se puedan añadir nuevas funcionalidades si se desea. El hecho de haber implementado el sistema en la plataforma de desarrollo de iPhone ha hecho mucho más sencillo dotar a la aplicación de una interfaz mucho más elaborada que lo que podría haber sido al utilizar por ejemplo Java. Pese a que en un principio el lenguaje de programación parecía muy similar a los vistos en otras ocasiones, como Java y C, las diferencias con respecto a estos son bastante notables. Como suele ocurrir, una vez familiarizado

con el lenguaje de Objective-C, el desarrollo se hace más eficientemente y con mayor detalle. Las herramientas ofrecidas, además, proporcionan una serie de recursos que hacen que la programación en este lenguaje sea muy agradable.

Hay que remarcar que para el lenguaje de Objective-C hay mucha documentación, tanto en formato electrónico como en papel, por lo que casi cualquier cosa que se pueda hacer, antes o después se termina consiguiendo.

7.2. Líneas futuras

En esta sección se proponen algunos cambios o mejoras que se podrían incluir en la aplicación.

1) Utilización de servidores: A través del uso de servidores, se podrían actualizar las puntuaciones de cada usuario casi en tiempo real, de manera que se podrían hacer públicas para que el resto de usuarios. Así un usuario podría ver las puntuaciones que tienen sus amigos (funcionamiento similar al que ofrece *Game Center* de Apple). A través del uso de servidores, la creación de un juego podría ser visible para todos los usuarios de la aplicación, no sólo aquel usuario al que se le envía el juego a través de Bluetooth.

2) Implementación en otras plataformas móviles: El proyecto actual ha sido implementado en la plataforma iPhone, pero sería interesante implementarlo también para otras plataformas tales como Android, que tienen una difusión también muy amplia, al ser plataformas de distribución gratuitas.

La implementación en una plataforma diferente implicaría realizar un esfuerzo similar al del presente proyecto.

3) Introducción de otro tipo de contenido docente: Para introducir un contenido docente diferente al de los algoritmos criptográficos, habría que modificar el sistema considerablemente, no su estructura, pero sí su contenido. La mayor parte de los juegos son específicos de criptografía, por tanto, no tendría sentido utilizar, por ejemplo, el juego de *Diffie-Hellman* dentro de una temática diferente. Otros juegos, como por ejemplo *Puzle* o *Pregunta/Respuesta* sí se podrían reutilizar muy fácilmente.

4) Implementación en otros idiomas: La implementación en otro idioma es una tarea que se podría realizar sin tener que modificar en gran medida la aplicación ya existente. Por lo que este cambio sería uno de los más viables de realizar a corto plazo.

8. Bibliografía y Glosario de Términos

8.1. Bibliografía

INTRODUCCIÓN

[1] **Canalys**

http://www.canalys.com/pr/2011/r2011051.html

Accedido en Mayo de 2011.

[2] **The Nielsen Company**

http://www.nielsen.com/content/corporate/global/en.html

Accedido en Mayo de 2011.

[3] **El País – Comparativa ventas Android con respecto a iPhone**

http://www.elpais.com/articulo/tecnologia/Android/supera/ventas/iPhone/Estados/Unidos/elpeputec/20100511elpeputec_1/Tes.

Accedido en Mayo de 2011.

[4] **ADSL Zone- Comparativa ventas de Nokia, Android y iPhone**

http://www.adslzone.net/article5907-nokia-y-apple-superan-a-android-en-europa.html

Accedido en Mayo de 2011.

[5] **Punto Geek- Sistemas operativos para móviles**

http://www.puntogeek.com/2010/06/08/5-sistemas-operativos-linux-para-telefonos-moviles/

Accedido en Mayo de 2011.

[6] **Gizmos – 5 mejores móviles con linux.**

http://www.gizmos.es/5187/moviles/5-mejores-moviles-con-linux/

Accedido en Mayo de 2011.

ESTADO DE LA CUESTIÓN

[7] **iTunes**

http://www.apple.com/es/itunes/

Accedido en Mayo de 2011.

[8] **Game Center**

http://www.apple.com/es/iphone/features/game-center.html

Accedido en Mayo de 2011.

[9] **Apple Store**

http://store.apple.com/es

Accedido en Mayo de 2011.

[10] **Cryptix Lite en iTunes**

http://itunes.apple.com/es/app/cryptix-lite/id301215132?mt=8Glosario

Accedido en Mayo de 2011.

[11] **Crostix Free en iTunes**

http://itunes.apple.com/es/app/crostix-free/id317783630?mt=8

Accedido en Mayo de 2011.

[12] **ARG Tool en iTunes**

http://itunes.apple.com/es/app/arg-tools/id374959430?mt=8

Accedido en Mayo de 2011.

[13] **Alicia Ramírez, criptografía**

http://www.aliciaramirez.com/crypto/index.html

Accedido en Mayo de 2011.

[14] **Great Day Games, juegos de criptografía**

http://www.greatdaygames.com/games/cryptogram/cryptogram.aspx

Accedido en Mayo de 2011.

[15] **Kriptópolis, juegos de criptografía y seguridad**

http://www.kriptopolis.org

Accedido en Mayo de 2011.

[16] **Titanes DC, juegos de criptografía y seguridad**

http://www.titanesdc.com/foros/matematicos-criptograficos-Puzles-juegos/

Accedido en Mayo de 2011.

[17] **Gaussianos, descifrado de criptogramas**

http://gaussianos.com/criptografia-descifrando-un-criptograma/

Accedido en Mayo de 2011.

[18] **Criptoblog, descifrado de criptogramas**

http://criptoblog.blogspot.com/

Accedido en Mayo de 2011.

GESTIÓN DEL PROYECTO

[19] **ESTÁNDARES DE INGENIERÍA DE SOFTWARE ESA - European Space Agency en versión Lite**

http://www.arcos.inf.uc3m.es/~ii_si/

Accedido en Mayo de 2011.

[20] **European Space Agency – ESA Software Engineering & Standardisation**

http://www.esa.int/TEC/Software_engineering_and_standardisation/

Accedido en Mayo de 2011.

ANÁLISIS DE COSTES

[21] **Anuncio laboral para probador**

http://lksite.superforo.net/t587-quieres-trabajar-de-tester-de-videojuegos

Accedido en Mayo de 2011.

[22] **Tablas de cotización de la seguridad social**

http://www.seg-social.es/Internet_1/Trabajadores

/CotizacionRecaudaci10777/Basesytiposdecotiza36537/index.htm

Accedido en Mayo de 2011.

[23] **Adobe- Photoshop**

http://www.adobe.com/support/downloads/product.jsp?product=39&platform=Macintosh

Accedido en Mayo de 2011.

[24] **Apple Store**

http://store.apple.com/es

Accedido en Mayo de 2011.

[25] **Microsoft Office for Mac**

http://www.microsoft.com/mac/products

Accedido en Mayo de 2011.

[26] **Altova**

http://www.altova.com/

Accedido en Mayo de 2011.

[27] **Top Ventas**

http://www.applesfera.com/curiosidades/otro-estudio-apunta-a-que-los-desarrolladores-tienen-mas-exito-en-ios

Accedido en Mayo de 2011.

[28] **Comisión para Apple de los beneficios de la venta de las aplicaciones.**

http://www.ticbeat.com/economia/desarrolladores-ganan-batalla-apple-comision/

Accedido en Mayo de 2011.

[29] **Introducing iAd Producer**

http://developer.apple.com/iad/iadproducer/

Accedido en Mayo de 2011.

ANÁLISIS

[30] **Cifrado César**

http://es.wikipedia.org/wiki/Cifrado_César

Accedido en Mayo de 2011.

[31] **Cifrado Vigenere**

http://es.wikipedia.org/wiki/Cifrado_de_Vigenère

Accedido en Mayo de 2011.

[32] **Cifrado Playfair**

http://es.wikipedia.org/wiki/Cifrado_de_Playfair

Accedido en Mayo de 2011.

[33] **Cifrado con Escítala**

http://es.wikipedia.org/wiki/Esc%C3%ADtala

Accedido en Mayo de 2011.

[34] **Protocolo de intercambio de claves Diffie-Hellman**

http://es.wikipedia.org/wiki/Diffie-Hellman

Accedido en Mayo de 2011.

- **Apuntes propios de Seguridad en las Tecnologías de la Información**

DISEÑO

[35] **MSDN- Diseño de Software**

http://msdn.microsoft.com/es-es/library/dd409436.aspx

Accedido en Mayo de 2011.

[36] **MSDN- Desarrollo de diagramas de componentes**

http://msdn.microsoft.com/es-es/library/dd409390.aspx

Accedido en Mayo de 2011.

IMPLEMENTACIÓN Y PRUEBAS

[37] **Lenguaje de programación Objective-C**

http://es.wikipedia.org/wiki/Objective-C

Accedido en Mayo de 2011.

[38] **The Objective-C programming language**

http://developer.apple.com/library/ios/#documentation/Cocoa/Conceptual/ObjectiveC/Introduction/introObjectiveC.html

Accedido en Mayo de 2011.

[39] **Stephen G.Kochan.** Programming in Objective-C: Sams Publishing, 2004.

[40] **Dave Mark, Jeff LaMarch.** More iPhone 3 Development: Appress, 2010.

[41] **Dave Mark, Jeff LaMarch, Jack Nutting.** Beginning iPhone 4 Development: Appress, 2011.

Glosario de Términos

[42] **SmartPhone**

http://es.wikipedia.org/wiki/Smartphone

Accedido en Mayo de 2011.

[43] **Algoritmo**

http://es.wikipedia.org/wiki/Algoritmo

Accedido en Mayo de 2011.

[44] **Aplicación Informática**

http://es.wikipedia.org/wiki/Aplicación_informática

Accedido en Mayo de 2011.

[45] **Metodología de Desarrollo del Software**

http://es.wikipedia.org/wiki/Metodolog%C3%ADa_de_desarrollo_de_software

Accedido en Mayo de 2011.

[46] **Ejecución**

http://es.wikipedia.org/wiki/Ejecución

Accedido en Mayo de 2011.

[47] **Backup**

http://es.wikipedia.org/wiki/Copia_de_seguridad

Accedido en Mayo de 2011.

[48] **Criptografía**

http://es.wikipedia.org/wiki/Criptograf%C3%ADa

Accedido en Mayo de 2011.

[49] **Implementar**

http://buscon.rae.es/draeI/SrvltConsulta?TIPO_BUS=3&LEMA=implement

Accedido en Mayo de 2011.

[50] **Interfaz**

http://buscon.rae.es/draeI/SrvltGUIBusUsual?LEMA=interfaz

Accedido en Mayo de 2011.

[51] **Framework**

http://es.wikipedia.org/wiki/Framework

Accedido en Mayo de 2011.

8.2. Glosario de Términos

SmartPhone: Término adquirido del inglés para referirse a un teléfono inteligente (teléfono móvil común que ofrece un mayor número de funcionalidades). Entre las funcionalidades ofrecidas se suele encontrar un cliente de correo electrónico. También permiten la instalación de programas que incrementan el procesamiento de datos y la conectividad [42].

Algoritmo: Es un conjunto predefinido de instrucciones o reglas bien definidas, ordenadas y finitas que permiten realizar una actividad mediante pasos sucesivos que no generan dudas a quien realiza la actividad. Partiendo de un estado inicial y una entrada, se llega a un estado final donde se obtiene una solución, siguiendo los pasos sucesivamente [43].

Aplicación (informática): Es un tipo de programa informático diseñado como herramienta para permitir a un usuario realizar uno o diversos tipos de trabajo [44].

Metodología (de desarrollo de software): Es un marco de trabajo usado para estructurar, planificar y controlar el proceso de desarrollo en sistemas de información [45].

Ejecutar: Poner a funcionar un programa [46].

Backup: Termino adquirido del inglés que quiere decir copia de seguridad. Estas copias de seguridad se hacen con el fin de que puedan utilizarse para restaurar la versión original después de haberse producido una eventual pérdida de datos [47].

Criptografía: Es la técnica utilizada, bien sea aplicada al arte o la ciencia, que altera las representaciones lingüísticas de un mensaje [48].

Implementar: Poner en funcionamiento, aplicar métodos, medidas, etc. para llevar algo a cabo [49].

Interfaz: Conexión física y funcional entre dos aparatos o sistemas independientes [50].

Framework: Referente al desarrollo del software, es una estructura conceptual y tecnológica de soporte definida, normalmente con artefactos o módulos de software concretos, con base en la cual otro [51].

Anexo 1. Manual de Usuario

A continuación se detallan todas las funcionalidades presentes en la aplicación y las cómo puede hacer uso de ellas el usuario.

Instalación

La aplicación *Enigmatium* requiere ser descargada de Apple Store (en el momento de escribir el presente documento la aplicación todavía no ha sido subida a Apple Store).

Una vez descargada la aplicación en el iPhone, iPad o iPod touch, el usuario ya puede acceder a ella a través del icono:

Imagen 44 – Icono de *Enigmatium*.

Menú Principal

Al comenzar la aplicación, aparece el menú principal (como se puede ver en la *Imagen 45*), a través del cual se puede acceder al proyecto de *Minijuegos*, al de *Aventura* y al menú de *Opciones*.

Imagen 45 – Interfaz Menú Principal.

Menú Opciones

Dentro del menú de opciones, como se puede ver en la *Imagen 46* el usuario puede acceder a los *Resultados* o al *Perfil*. Si el perfil de usuario no está creado, el usuario no podrá comenzar a jugar a ninguno de los minijuegos ya que no podrá acceder al menú de minijuegos.

Imagen 46 – Interfaz Menú Opciones.

Menú Perfil

Dentro de *Perfil*, la primera vez que acceda el usuario tendrá que crear su propio perfil, con los siguientes datos propios: *alias*, *nombre*, *apellidos*, *correo electrónico* e *imagen*. Antes de introducir ningún dato, el aspecto de la pantalla será el que se muestra en la *Imagen 47.* Después de introducir los datos personales, la interfaz quedará como se muestra en la *Imagen 48*. Con todos los campos completos, ya se puede guardar el perfil (pulsando el botón de *Guardar*).

Para modificar el perfil, el usuario debe modificar el campo o los campos que considere necesario (la interfaz que se le muestra es como la *Imagen 48*), y después debe guardar esos nuevos datos (pulsando el botón de G*uardar*).

Para eliminar el perfil, el usuario debe borrar todos los campos y posteriormente pulsar el botón de *Guardar*.

Imagen 47 – Interfaz Perfil vacío.

Imagen 48 – Interfaz Perfil con Datos Usuario.

Después de pulsar el botón de *Guardar*, se le pregunta al usuario si realmente quiere guardarlo (ver *Imagen 49*).

Imagen 49 – Interfaz Guardar Usuario.

Si el usuario selecciona guardar usuario, se le mostrará un mensaje de información indicándolo (como el que se muestra en la *Imagen 50*).

Imagen 50 – Interfaz Usuario Guardado.

Menú Resultados

Si el usuario accede al menú de resultados podrá comprobar los puntos que ha obtenido en los diferentes juegos. Cada uno de los juegos está dividido por sus correspondientes niveles. En la parte superior de la pantalla aparecen los puntos totales, es decir, la suma de las puntuaciones obtenidas en cada uno de los niveles. La interfaz del menú puntuaciones se muestra en la *Imagen 51.*

Imagen 51 – Interfaz Usuario Guardado.

Minijuegos

En la primera pantalla del menú de minijuegos, el usuario puede elegir o bien jugar a alguno de los juegos que vienen por defecto en la aplicación, o bien crear algún juego nuevo. Esta interfaz se ve en la I*magen 52.*

Imagen 52 – Interfaz Menú Minijuegos.

Menú Crear

Aunque el usuario puede perfectamente acceder al menú de *Crear*, no es competencia de este proyecto realizar a explicación de los elementos que se encuentran en su interior. El usuario debe dirigirse al manual de usuario del proyecto *Enigmatium-Aventura* para continuar las explicaciones de lo que puede hacer.

Menú Jugar

Al acceder al menú *Jugar* el usuario se encuentra con un nuevo menú en el que podrá elegir si acceder a su vez al menú de *Juegos Propios* (estos son los juegos que vienen por defecto con la aplicación) o bien acceder al menú de *Juegos Amigos*. Esta interfaz se muestra en la I*magen 53*.

Imagen 53 – Interfaz Menú Jugar.

Menú Juegos Amigos

Aunque el usuario puede perfectamente acceder al menú de *Juegos Amigos*, no es competencia de este proyecto realizar a explicación de los elementos que se encuentran en su interior. El usuario debe dirigirse al manual de usuario del proyecto *Enigmatium-Aventura* para continuar las explicaciones de lo que puede hacer.

Menú Minijuegos

Desde la interfaz de *Minijuegos* el usuario puede acceder a cualquiera de los 7 juegos que contiene la aplicación: *Puzle, Escítala, César, Playfair, Vigenere, Diffie-Hellman o Pregunta/Respuesta*. Este último juego de *Pregunta/Respuesta*, no está a simple vista, el usuario debe desplazar un dedo sobre la pantalla del dispositivo con un movimiento de derecha a izquierda, el mismo movimiento que se hace al pasar una página de un libro. Entonces se mostrará el último de los minijuegos.

La primera interfaz con los juegos se puede ver en la *Imagen 54*. La segunda y tercera de las imágenes muestra la transición entre la primera interfaz y la segunda, donde se encuentra el último de los juegos.

Imagen 54 – Interfaz Menú Juegos.

Elementos comunes

En todos los juegos, los elementos mostrados en la *Imagen 55* son comunes, y por lo tanto se explican una vez, pero para el resto de juegos funcionan igual.

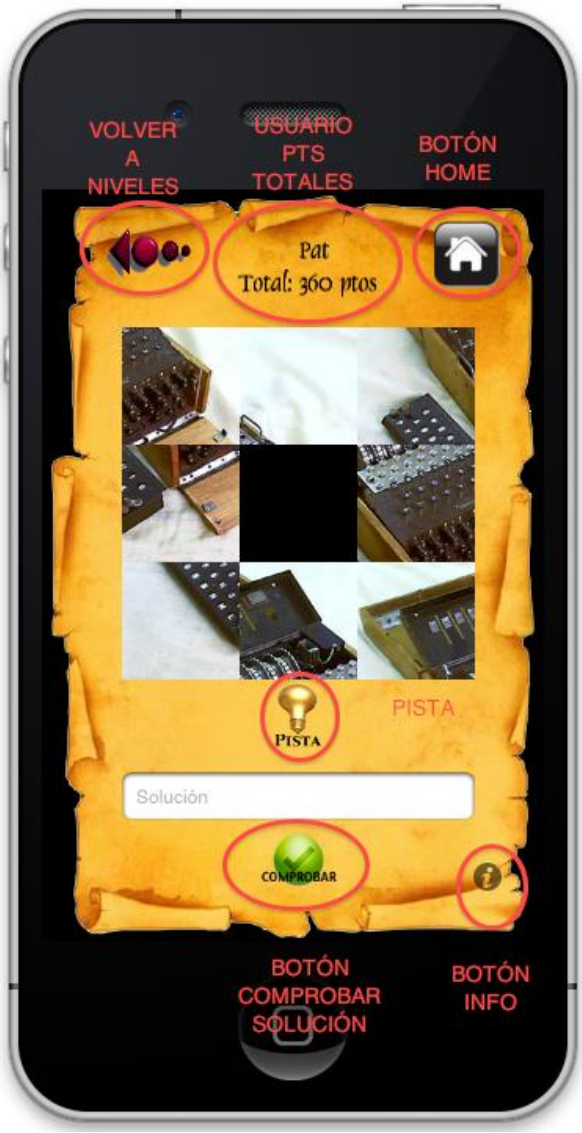


Imagen 55 – Interfaz Botones Comunes.

1) Volver a niveles: Si se pulsa en el botón situado en la esquina superior izquierda vuelve a la interfaz con los niveles de los juegos (en este caso retorna a los niveles de *Puzle*).

2) Botón Home: Si se pulsa en el botón situado en la esquina superior derecha, aparece el menú *Home,* que se explicará más adelante.

3) Usuario y Puntos Totales: La información que se muestra en la parte superior, entre los botones de *Volver* y *Home* se corresponden con el *Alias* del usuario y con los puntos totales que este usuario lleva acumulados.

4) Botón Info: En la esquina inferior derecha se encuentra el botón con la información referente al juego que se muestra.

5) Botón Comprobar Solución: El botón que se encuentra en la parte central inferior se utiliza para comprobar si la solución introducida al juego es correcta.

Menú Home

En este menú, el usuario puede acceder a los botones que se muestran en la *Imagen 56.*

Imagen 56 – Interfaz Menú Home.

1) Continuar: Lleva al usuario de vuelva al juego donde se encontraba.

2) Resultados: Lleva al usuario al menú *Resultados* donde puede consultar los puntos conseguidos en cada juego.

3) Minijuegos: Lleva al usuario al menú de *Minijuegos.*

4) Menú Inicio: Lleva al usuario al menú de *Inicio*.

Sistema de Puntuaciones

La primera vez que el usuario resuelve el juego se le otorga una puntuación máxima de 100 puntos. Si el juego se resuelve en el segundo intento, se le da al usuario una puntuación de 80 puntos. Si lo resuelve al tercer intento, la puntuación obtenida será de 60 puntos. En la intento número cuatro, se le dará una puntuación de 40 puntos. Al resolverlo en el quinto intento, se le dará al usuario una puntuación de 20 puntos. Si el usuario resuelve el juego en el número de intento seis o posterior sólo se le darán 10 puntos, sea cual sea el número de intento posterior al sexto que sea.

Si el usuario resuelve de manera correcta el Puzle, en la pantalla de niveles, le aparecerán los puntos conseguidos en dicho nivel, y a su vez se desbloqueará el siguiente nivel. De esta manera el usuario puede volver a jugar tantas veces como quiera al juego que acaba de superar así como al siguiente nivel del juego. Todos los niveles que hayan sido superados no volverán a dar puntuaciones al usuario. La interfaz de usuarios se mostrará como en la *Imagen 57*.

Imagen 57 – Interfaz Niveles Superados.

Juego Puzle

Lo primero que aparece es la pantalla con los diferentes niveles del juego de Puzle. La primera vez que el usuario acceda a esta pantalla, como no ha superado ningún nivel, sólo puede acceder al nivel 1 y hasta que no lo supere, no se desbloqueará el nivel 2.

En la *imagen 58* se puede observar el aspecto de la interfaz de los niveles de Puzle.

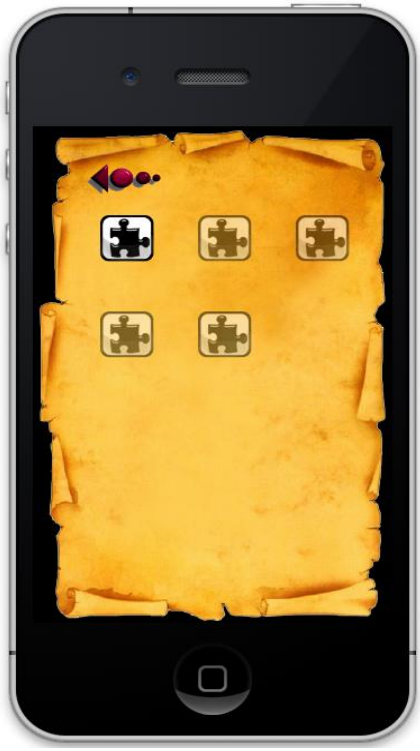


Imagen 58 – Interfaz Niveles Puzle.

Al acceder al juego, el usuario ya puede empezar a jugar al juego de Puzle. La interfaz del juego es como se muestra en la *Imagen 59.*

Imagen 59 – Interfaz Juego Puzle.

El usuario mueve las casillas un número indeterminado de veces hasta que obtenga una imagen clara y bien definida de la fotografía que se encuentra escondida bajo los fragmentos que se ven en la *Imagen 59*. Para mover las casillas, el usuario tendrá que pulsar sobre un fragmento de imagen que se encuentre contiguo al fragmento vacío. Los movimientos de las imágenes sólo se realizan en horizontal y vertical, pero no en diagonal.

Cuando la imagen está bien colocada, el usuario puede comprobar la solución al juego, aunque si no ha introducido una respuesta escrita de lo que le sugiere la imagen, en caso de estar correctamente colocada, se le darán menos puntos. Si además de colocar correctamente la imagen, el usuario introduce también de manera correcta la solución, se le darán los puntos que le correspondan con el número de intento en el que se encuentre.

Si el usuario necesita una pista sobre el concepto que representa la imagen, puede pulsar el botón de pista, aunque perderá 10 puntos.

En la *imagen 60* se muestra la información relativa al juego de *Puzle* que obtiene el usuario cuando pulsa en botón de *información.*

Imagen 60 – Interfaz Información Puzle.

Juego Escítala

En la *Imagen 61* se muestra la interfaz relativa al juego de *Escítala.*

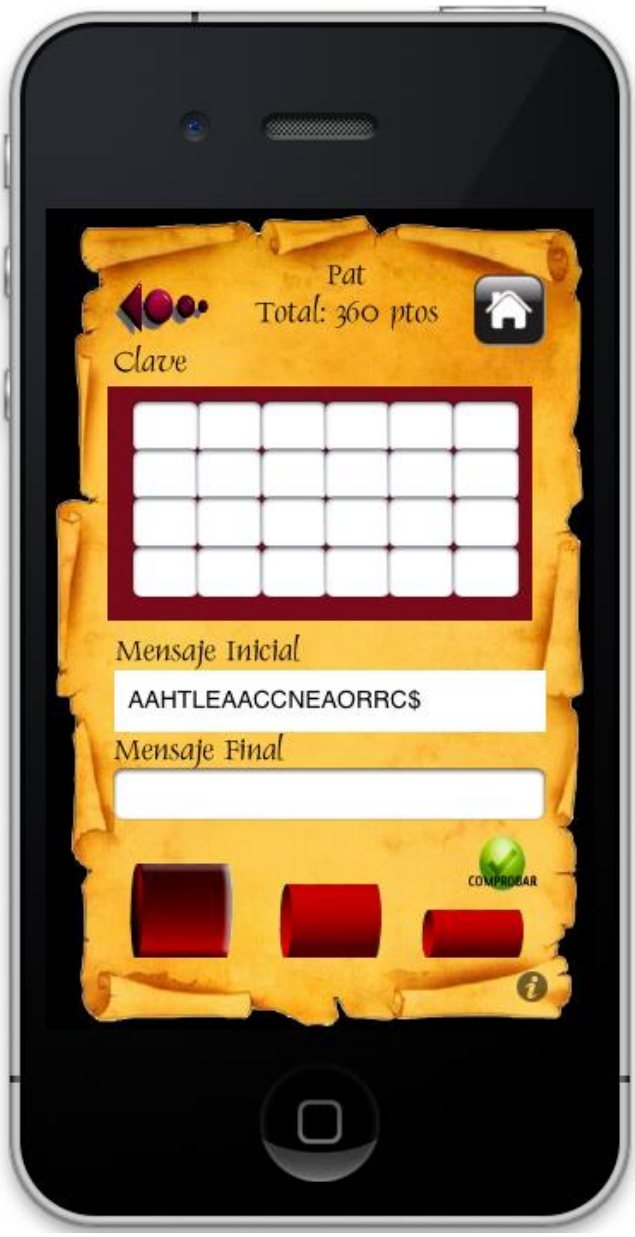


Imagen 61 – Interfaz Juego Escítala.

El usuario debe colocar el mensaje inicial en la matriz donde pone *Clave*. El mensaje se coloca de manera vertical en la matriz y se lee de manera horizontal para obtener el mensaje final. La colocación del mensaje inicial se muestra en la *Imagen 62*.

Imagen 62 – Interfaz Juego Escítala Incorrecto.

Imagen 63 – Interfaz Información Puzle.

Como se puede ver en la *Imagen 62*, al terminar de colocar el mensaje inicial, sobran casillas en la matriz, lo que significa que no se ha elegido el tamaño de escítala correcto. Esto se soluciona cambiando el tamaño de la escítala (como se puede ver en la *Imagen 63*) de manera que todas las casillas de la matriz queden completas y se pueda leer el mensaje final en horizontal dentro de la matriz.

El tamaño de la escítala se puede cambiar eligiendo uno de los tres tamaños que se encuentran en la parte inferior de la pantalla.

Juego César

En la *imagen 65* se muestra la pantalla inicial del juego de César. En ella aparece el desplazamiento el usuario debe seleccionar para poder resolver el juego.

Una vez que retira la nube de información con el desplazamiento, le aparece al usuario la pantalla principal (esto se puede ver en la *Imagen 64* y en la *Imagen 65*). Con ayuda de la ruleta, el usuario debe identificar cuál es la letra en claro a partir de la letra cifrada que tiene en el mensaje cifrado.

Cuando tenga todo el mensaje en claro resuelto, debe comprobar si la solución introducida es correcta.

Imagen 64 – Interfaz Pista César.

Imagen 65 – Interfaz Juego César.

Juego Playfair

En la *Imagen 67* se muestra la pantalla inicial del juego de *Playfair*. En ella aparece la clave que el usuario debe seleccionar de la tabla de claves para poder resolver el juego.

Una vez que retira la nube de información con la clave a utilizar, le aparece al usuario la pantalla principal (esto se puede ver en la *Imagen 66* y en la *Imagen 67*).

Imagen 66 – Interfaz Pista Playfair.

Imagen 67 – Interfaz Pista Playfair.

Pulsando en el botón de *Configurar Clave*, accede a la pantalla que se muestra en la *Imagen 68*. Y pulsando al botón de *Seleccionar Clave*, el usuario podrá elegir una clave de las que se muestran en la *Imagen 69*.

Imagen 68 – Interfaz Pista Playfair.

Imagen 69 – Interfaz Pista Playfair.

Al seleccionar la clave, ésta se le colocará automáticamente en la matriz, dependiendo de si elige una colocación normal (*Imagen 70*) o en espiral (*Imagen 71*).

Imagen 70 – Interfaz Playfair Normal.

Imagen 71 – Interfaz Playfair Espiral.

Con ayuda de la ruleta, el usuario debe identificar cuál es la letra en claro a partir de la letra cifrada que tiene en el mensaje cifrado.

El usuario debe colocar el resto de las letras del abecedario siguiendo la misma colocación que ha seguido para la matriz. Pulsando al botón de volver, el usuario vuelve a la primera pantalla donde se encuentra el mensaje cifrado. La colocación del resto de la matriz la puede realizar en cualquiera de las dos pantallas.

Para resolver el mensaje cifrado, el usuario debe seguir el algoritmo de *Playfair* (que se encuentra explicado en el botón de información de la pantalla principal del juego de *Playfair*). Cuando el mensaje cifrado esté resuelto y el

usuario tenga escrito en la casilla del mensaje en claro el mensaje resuelto, podrá comprobar la solución al juego.

Cuando tenga todo el mensaje en claro resuelto, debe comprobar si la solución introducida es correcta.

Juego Vigenere

En la *imagen 73* se muestra la pantalla inicial del juego de *Vigenere*. En ella aparece la clave que el usuario debe seleccionar de la tabla de claves para poder resolver el juego.

Una vez que retira la nube de información con la clave a utilizar, le aparece al usuario la pantalla principal (esto se puede ver en la *Imagen 72* y en la *Imagen 73*).

Imagen 72 – Interfaz Vigenere Pista.

Imagen 73 – Interfaz Vigenere.

El usuario puede seleccionar la colocación de la clave, con clave o con autoclave. A continuación, al pulsar el botón de *Lista Clave* le aparece la misma tabla con las mismas claves guardadas que se ha utilizado para el juego de *Playfair* (esta interfaz no se va a mostrar).

Una vez que se selecciona la clave, se coloca en las casillas de *Clave Seleccionada*, como se ve en la *Imagen 74.*

Imagen 74 – Interfaz Vigenere con Clave.

Si el usuario pulsa al botón de *Continuar*, puede comenzar a jugar. Con la ayuda de las ruletas proporcionadas al usuario, éste debe coger una letra cifrada con una letra de la clave, colocar cada una en su correspondiente ruleta, y calcular la letra en claro que se corresponde. Los elementos de esta interfaz se muestran en la *Imagen 75*.

Imagen 75 – Interfaz Juego Vigenere.

Juego Diffie-Hellman

En la *imagen 76* se muestra la pantalla inicial del juego de *Diffie-Hellman*. En ella aparecen los datos que se le dan por defecto al usuario. El usuario debe introducir su clave privada, y calcular su clave pública.

Al darle al botón de continuar, le aparece la clave pública de la persona con la que va a calcular la clave compartida. Una vez la calcule dándole al botón correspondiente, debe resolver el mensaje que se encuentra cifrado pulsando al correspondiente botón. Puede comprobar la solución obtenida mediante el botón de *Comprobar*. Todas estas funcionalidades se muestran en la *Imagen 76* y en la *Imagen 77.*

Imagen 76 – Interfaz Juego Diffie-Hellman.

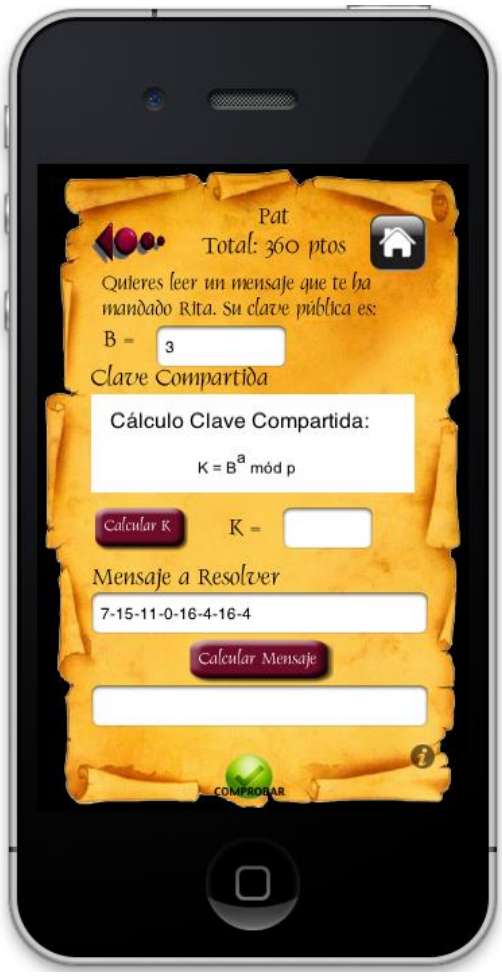


Imagen 77 – Interfaz Juego Diffie-Hellman 2.

Juego Pregunta/ Respuesta

Cuando el usuario entra a jugar a un nivel del juego *Pregunta/Respuesta* puede jugar con dos tipos de configuraciones diferentes: Trivial + Tabú o bien Pictionary + Contador.

Con la primera configuración, *Trivial + Tabú*, la primera de las interfaces que salen es la que se muestra en la *Imagen 78.*

Imagen 78 – Interfaz Juego Trivial.

Cuando el usuario supera este juego, el juego de *Tabú* se le muestra al usuario (ver *Imagen 79*).

Imagen 79 – Interfaz Juego Trivial.

Para superar el nivel donde sale esta configuración, se deben superar los dos juegos, no es suficiente con superar únicamente el juego de *Trivial*.

El usuario también puede encontrarse en un nivel con los juegos de *Pictionary* y *Contador*. Estas interfaces se muestran en la *Imagen 80* y la *Imagen 81.*

Imagen 80 – Interfaz Juego Pictionary.

Imagen 81 – Interfaz Juego Contador.

Cuando el usuario supera el juego de *Pictionary* automáticamente pasa al juego de *Contador*, donde se le avisa que tiene 60 segundos para completar todas las respuestas del juego. Cuando finalizan esos 60 segundos, no puede seguir escribiendo. Se suman los puntos sólo cuando se superan los dos juegos.

La interfaz donde tiene que introducir los resultados de *Contador* se puede ver en la *Imagen 82*.

Imagen 82 – Interfaz Juego Contador 2.

Anexo 2. Test de evaluación por parte del usuario final

ENIGMATIUM

Valora del 1 al 10, siendo 1 la menor puntuación y 10 la mayor.

1) ¿Te parece intuitivo el sistema?

1 2 3 4 5 6 7 8 9 10

2) ¿Has podido jugar a todos los juegos?

SÍ NO

3) Si has contestado no a la anterior, ¿a qué juego no has podido jugar?

Puzle Vigenere César Escitala Playfair Diffie-Hellman Pregun-ta/Respuesta

4) ¿Has sabido interpretar todos los botones que aparecen en la aplicación?

1 2 3 4 5 6 7 8 9 10

5) ¿Los botones de ayuda te han resultado útiles? Valora su utilidad

1 2 3 4 5 6 7 8 9 10

6) ¿Ha habido algún momento en el que no sabías por dónde seguir, donde te hayas quedado atascado/a?

1 2 3 4 5 6 7 8 9 10

7) ¿Te parece útil la aplicación para reforzar tus conocimientos acerca de los algoritmos criptográficos y de seguridad? Valora su utilidad

1 2 3 4 5 6 7 8 9 10

8) Después de haber usado la aplicación, si tuvieras que pagar por ella para descargártela, ¿estarías dispuesto/a?

SÍ NO

9) ¿Qué valoración general le darías a la aplicación?

1 2 3 4 5 6 7 8 9 10

(Por favor, rellena estos apartados, así nos ayudas a mejorar nuestro proyecto!!)

10) Cosas a mejorar.

11) Puntos fuertes que has encontrado.

12) Comenta lo que quieras acerca de la aplicación.